



添付公開書類:
— 国際調査報告書

2文字コード及び他の略語については、定期発行される各PCTガゼットの巻頭に掲載されている「コードと略語のガイダンスノート」を参照。

明 細 書

演算処理装置、その構築方法、演算処理システム及び演算処理方法

5 技術分野

本発明は、システムプログラムに基づいて各種データ処理をする中央演算装置（CPU）やマイクロプロセッサユニット（MPU）等、また、プログラム可能な論理演算素子（PLD）や、これらの組み込み電子機器等に内蔵可能でプログラム可能な1チップマイクロコンピュータ等に適用して好適な演算処理装置、その構築方法、演算処理システム及び演算処理方法に関するものである。

詳しくは、演算命令を実行するためのレジスタアレイ及び演算部を制御する命令実行制御部を備え、演算命令に基づいて一のレジスタを選択すると共に、このレジスタが保持する値によって他のレジスタを選択するレジスタ相対レジスタアドレッシング処理を実行して、レジスタアレイをデータの随時書込み及び読出し可能なメモリのように動作できるようにすると共に、CPUや、RAM、ROM等を個々に基板に配置する場合に比べて基板占有面積を低減できるようにしたものである。

この発明では、複数のレジスタを指定して任意の演算を実行する演算処理装置を備え、プログラム作成系で作成された圧縮プログラムを取得してレジスタ種類を解読し、このレジスタ種類に基づいて当該レジスタを指定する命令ビット数を復元し、所定の命令長の命令構造を有するプログラムを復元するようにして、プログラムデータを格納するROM等のメモリ容量を低減できるようにすると共に、メモリセルや論理演算素子から成るPLDによりプロセッサを構築する場合に、ROMとして機能させるメモリセルの占有率を低減できるようにしたものである。

25

背景技術

近年、携帯端末装置や、電子カード、情報処理装置等の各種電子機器にCPU（中央演算処理装置）を含むマイクロプロセッサが使用される場合が多くなってきた。この種のプロセッサで記憶装置にアクセスしようとした場合、アクセス先

がレジスタである場合と外部メモリである場合とで別々のアクセス方法が採られる。アクセス先がレジスタである場合、例えば、ALU (Arithmetic Logic Unit: 算術論理演算ユニット) の周辺にはコピーレジスタやテンポラリレジスタ等が配置され、演算処理を行う場合、コピーレジスタにデータを複写したり、演算処理後のデータをテンポラリレジスタに一時記憶するようになされる。ALUがレジスタ対レジスタの演算処理に適しているためである。アクセス先が外部メモリである場合、例えば、当該レジスタ番号が示す外部メモリの番地に情報を書込む場合、CPUが当該外部メモリに対して、その格納番地を指定する書込みアドレスと、書き込み信号とを出力する。当該レジスタ番号が示す外部メモリの番地から情報を読み出す場合は、CPUが当該外部メモリに対して、その格納番地を指定する読出しアドレスと、読出し信号を出力するようになされる。

このように、当該レジスタ番号が示す外部メモリの番地に情報を格納したり、当該レジスタ番号が示す外部メモリの番地から情報を読み出す際に、CPUが当該外部メモリの格納番地（書込みアドレス又は読出しアドレス）を指定する。このような処理はレジスタ相対メモリアドレッシングと呼ばれる場合が多い。これは、通常、外部メモリがプロセッサと異なるデバイスに実装されているためである。

また、この種のプロセッサには命令実行演算部の他に命令実行プログラムを格納するための読み出し専用メモリ（以下、ROMという）や、命令実行演算処理に使用される多くのレジスタ等が実装されている。従来方式のマイクロプロセッサによれば、任意の電子機器に当該プロセッサを組み込んで用いる場合、ある動作を行う命令はその動作と、その動作を行うための命令とが一對一に対応していた。つまり、使用頻度が高いレジスタも、使用頻度が低いレジスタも、当該レジスタを指定するための命令ビット数を同等にして一律の長さの命令により作成されたプログラムが使用される場合が多い。このことで、固定長の命令がROMに格納されて使用される。

他方で、半導体集積回路技術の発展により、多大な数のレジスタをプロセッサ（以下演算処理装置ともいう）内に実装することが可能になってきた。この場合、レジスタを特定するための命令ビット数も益々多く必要になる。例えば、102

4 個のレジスタが実装される場合、1024 個のレジスタ中でその 1 つを特定するためには、命令ビットとして 10 ビットが必要となる。しかし、実際のプログラムでは全てのレジスタへのアクセス頻度は一様ではなく、アクセス頻度に斑がある。頻繁にアクセスされるレジスタの番号は一般的にコンパイラによって決定
5 される。

ところで、従来方式の演算処理装置によれば、以下の理由により、処理速度の低下を招いていた。

① 外部メモリにアクセスするためのアクセス時間がかかることである。これは一般的に記憶装置は特定のプロセッサに特化して作られていることは希である
10 ことから、プロセッサ側で、アクセスをしたい記憶装置に合わせたインターフェースを用意しなければならない。従って、記憶装置にアクセスする度にこのインターフェースを経由するため、アクセス時間がかかり、演算処理速度が低下する。

② 外部メモリから ALU へ、また ALU から外部メモリへデータを転送する際の時間がかかる。これは一般的に ALU がレジスタ対レジスタの演算処理に対応しているため、外部メモリ上に演算したいデータがある場合、一旦、外部メモリからコピーレジスタへデータが複写され、その後、コピーレジスタから ALU
15 へデータが転送されて演算処理を行うようになされる（レジスタ相対メモリアドレスリング）。従って、演算処理速度が低下する原因となる。

③ CPU や、RAM、ROM 等の機能を 1 チップに集合化して 1 チップマイクロコンピュータ等を構成しようとした場合に、同一半導体チップ上に CPU を配置し、その周辺部に RAM や ROM 等を配置する方法が考えられる。この方法は、レジスタ相対メモリアドレスリング処理に依存することとなって、演算処理速度の向上が見込めない。
20

④ また、従来方式の演算処理装置によれば、命令実行プログラムを格納する ROM を実装するに当たり以下のような問題がある。
25

この命令実行プログラムのコード全体を眺めてみると、レジスタを指定するための例えば、10 ビットの命令ビット内、この 10 ビット全体が一様に使用されることが少ない。従って、命令実行プログラムを格納するメモリ中（例えば ROM やフラッシュメモリ）に無駄なビットが多く存在してしまう。これにより、全

てのレジスタを単一の命令ビット数で表現する方法は効率良くROMを使用することに関して妨げとなる。

- ⑤ 更にまた、メモリセルや論理演算素子から成るプログラマブル・ロジック・デバイス (Programmable Logic Device ; PLD) によりマイクロプロセッサ等を構築しようとした場合に、同一半導体チップ上に命令実行演算部を配置し、その周辺部にレジスタアレイや、ROM等を配置する方法が考えられる。この場合に、プロセッサの多機能化の要求から命令実行プログラムが多くなると、このプログラムを格納するROMのメモリ容量が多く必要になる。従って、メモリセルがROM構築に占有されてしまい、多くのメモリセルをレジスタに割り当てる
- 10 ことが困難になる。

発明の開示

- 本発明に係る第1の演算処理装置は、任意の演算プログラムに基づいて演算処理をする装置である。この演算処理装置には、書込みアドレス及び書込み制御信号に基づいて任意の値を保持し、及び、読出しアドレスに基づいて当該値を出力するレジスタを複数有したレジスタアレイと、このレジスタアレイから読み出された値を演算する演算部と、この演算部を動作させるための演算プログラムから演算命令を解釈する命令解釈部と、この命令解釈部によって解釈された演算命令を実行するためにレジスタアレイ及び演算部を制御する命令実行制御部とが備えられる。この命令実行制御部は演算命令に基づいて一のレジスタを選択し、ここで選択されたレジスタが保持する値によって他のレジスタを選択するレジスタ相対レジスタアドレッシング処理を実行するものである。
- 15 20

- この第1の演算処理装置によれば、任意の演算プログラムに基づいて演算処理をする場合に、命令解釈部では演算部を動作させるための演算プログラムから演算命令が解釈される。命令解釈部によって解釈された演算命令を実行するために命令実行制御部ではレジスタアレイ及び演算部を制御する。これを前提にして、命令実行制御部では、演算命令に基づいて一のレジスタを選択し、ここで選択されたレジスタが保持する値によって他のレジスタを選択するレジスタ相対レジスタアドレッシング処理を実行するようになされる。レジスタアレイの個々のレジ
- 25

スタではレジスタ相対レジスタアドレッシング処理によって指定される書込みアドレス及び書込み制御信号に基づいて任意の値を保持し、及び、読出しアドレスに基づいて当該値を出力するようになされる。演算部ではレジスタアレイから読み出された値を演算するようになされる。

- 5 従って、レジスタアレイをデータの随時書込み及び読出し可能なメモリ（RAM）のように取り扱うことができるので、従来方式の中央演算処理装置（CPU）に比べて外部メモリの格納番地を指定するレジスタ相対メモリアドレッシング処理に依存しなくなる。従来方式のCPUの中にRAMやROMの機能を取り込むことができるので、高速演算処理を実行すること、及び、CPUや、RAM、
- 10 ROM等を個々に基板に配置する場合に比べて基板占有面積を低減することができる。当該演算処理装置が適用される応用装置のコンパクト化を図ることができる。しかも、同一半導体チップ上に、レジスタアレイや読出し専用メモリをメモリセルにより構成し、演算部、命令解読部及び命令実行制御部を算術論理演算素子により構成することで、当該演算処理装置をプログラマブル・ロジック・デバイスによって構成することができる。

- 本発明に係る各々の演算処理装置の構築方法は、任意の演算プログラムに基づいて演算処理をする装置を構築する方法である。この構築方法では、予め同一半導体チップ上に複数のメモリセル及び算術論理演算素子を形成し、メモリセルを組み合わせてレジスタアレイ及び読出し専用メモリを画定すると共に、算術論理
- 20 演算素子を組み合わせて演算部、命令解読部及び命令実行制御部を画定し、その後、レジスタアレイ、読出し専用メモリ、演算部、命令解読部及び命令実行制御部を予め設定された配線情報に基づいて結線すると共に、読出し専用メモリに任意の演算プログラムを書込むようになされる。

- 本発明に係る演算処理装置の構築方法によれば、例えば、電源オンと共に不揮
- 25 発性の記憶部から読み出された配線情報に基づいてレジスタアレイ、読出し専用メモリ、演算部、命令解読部及び命令実行制御部を結線するようになされる。従って、命令実行制御部で演算命令に基づいて一のレジスタを選択し、このレジスタが保持する値によって他のレジスタを選択するレジスタ相対レジスタアドレッシング処理を実行するようなプログラム可能な演算処理装置を電源オンと共に構

築することができる。しかも、従来方式のCPUの中にRAMやROMの機能を取り込んだ演算処理装置を構築することができ、高速演算処理を実行すること、及び、CPUや、RAM、ROM等を個々に基板に配置する場合に比べて基板占有面積を低減することができる。当該演算処理装置が適用される応用装置のコンパクト化を図ることができる。

本発明に係る第1の演算処理方法は、演算プログラムに基づいて任意の演算処理をする方法である。この演算処理方法では、書込みアドレス及び書込み制御信号に基づいて任意の値を保持し、及び、読出しアドレスに基づいて当該値を出力するレジスタを複数準備し、その後、演算プログラムから演算命令を解読し、演算命令に基づいて一のレジスタを選択し、ここで選択されたレジスタが保持する値によって他のレジスタを選択するレジスタ相対レジスタアドレッシング処理を実行すると共に、演算命令に基づいて他のレジスタを選択し、ここで選択された他のレジスタが保持する値とレジスタ相対レジスタアドレッシング処理によって選択されたレジスタの値とを演算するようになされる。

この第1の演算処理方法によれば、演算プログラムに基づいて任意の演算処理をする場合に、複数のレジスタをデータの随時書込み及び読出し可能なメモリのように取り扱うことができるので、従来方式の中央演算処理装置と外部メモリと組み合わせた演算処理方法に比べてレジスタ相対メモリアドレッシング処理に依存しなくなる。これにより、従来方式に比べて高速演算処理を実行できるようになる。

本発明に係る第2の演算処理装置は、レジスタ相対レジスタアドレッシング処理を実行する演算プログラムに基づいて演算処理をする装置であって、複数のレジスタと、レジスタを使用する頻度に基づいて当該レジスタを指定する命令ビット数が予め圧縮されると共に、前記プログラムの命令構造の中にレジスタ種類が記述された命令長の異なる圧縮プログラムを記憶する記憶部と、この記憶部から圧縮プログラムを読み出してレジスタ種類を解読し、当該レジスタ種類に基づいてレジスタを指定するための命令ビット数を復元する命令解読部と、この命令解読部によって復元された所定長さの命令に基づいてレジスタを指定して任意の演算を実行する命令実行演算部とを備えるものである。

- この第2の演算処理装置によれば、レジスタを使用する頻度に基づいて当該レジスタを指定する命令ビット数が予め圧縮されると共に、当該プログラムの命令構造の中にレジスタ種類が記述された命令長の異なる圧縮プログラムに基づいてレジスタ相対レジスタアドレッシング処理を実行する場合に、記憶部には複数の
- 5 レジスタ中から当該レジスタを指定するための圧縮プログラムが記憶される。命令解読部では、この記憶部から圧縮プログラムを読み出してレジスタ種類が解読され、このレジスタ種類に基づいて当該レジスタを指定するための命令ビット数が復元される。これを前提にして、命令実行演算部では命令解読部によって復元された所定の命令長のプログラムに基づいてレジスタを指定して任意の演算を実
- 10 行するようになされる。従って、レジスタの使用頻度に応じて可変された命令の長さの圧縮プログラムであって、頻繁にアクセスするレジスタには短い長さの命令がセットされた、圧縮プログラムデータを記憶部に格納することができるので、そのメモリ容量を低減することができる。これにより、メモリセルや論理演算素子から成るPLDによりプロセッサを構築する場合に、ROMとして機能させる
- 15 メモリセルの占有率を低減することができ、その分のメモリセルをレジスタに多く割り当てることができる。

- 本発明に係る第2の演算処理方法は、レジスタ相対レジスタアドレッシング処理を実行する演算プログラムに基づいて演算処理をする方法である。この演算処理方法では、レジスタを使用する頻度に基づいて、当該レジスタを指定する命令
- 20 ビット数が予め圧縮される共に、プログラムの命令構造の中にレジスタ種類が記述された命令長の異なる圧縮プログラムを記憶し、この圧縮プログラムを読み出してレジスタ種類を解読し、当該レジスタ種類に基づいてレジスタを指定するための命令ビット数を復元し、ここで復元された所定長さの命令に基づいてレジスタを指定して任意の演算を実行するようになされる。

- 25 この第2の演算処理方法によれば、レジスタ相対レジスタアドレッシング処理を実行する演算プログラムに基づいて演算処理をする場合に、レジスタの使用頻度に応じて可変された命令の長さの圧縮プログラムであって、頻繁にアクセスするレジスタには短い長さの命令がセットされた、圧縮プログラムデータをメモリ領域に格納することができるので、そのメモリ容量を低減することができる。こ

れにより、メモリセルや論理演算素子から成るPLDによりプロセッサを構築する場合に、ROMとして機能させるメモリセルの占有率を低減することができ、その分のメモリセルをレジスタに多く割り当てることができる。

- 本発明に係る演算処理システムは、一方で、所定のプログラム言語に基づいて
- 5 レジスタ相対レジスタアドレッシング演算を実行するための命令を編集してプログラムを作成するプログラム作成装置と、他方で、当該プログラムと複数のレジスタとを使用してレジスタ相対レジスタアドレッシング演算を実行する演算処理装置とを有する演算処理システムである。このプログラム作成装置は、レジスタを使用する頻度に基づいて当該レジスタを指定する命令ビット数を圧縮すると共に、
- 10 当該プログラムの命令構造の中にレジスタ種類を記述して命令長の異なる圧縮プログラムを作成する。この演算処理装置は、プログラム作成装置で作成された圧縮プログラムを取得してレジスタ種類を解読し、レジスタ種類に基づいて当該レジスタを指定する命令ビット数を復元し、所定長さの命令に基づいて複数のレジスタを指定して任意の演算を実行するものである。
- 15 この演算処理システムによれば、一方で、所定のプログラム言語に基づいてレジスタ相対レジスタアドレッシング演算を実行するための命令を編集してプログラムを作成し、他方で、当該プログラムと複数のレジスタとを使用してレジスタ相対レジスタアドレッシング演算を実行する場合に、プログラム作成装置ではレジスタを使用する頻度に基づいて当該レジスタを指定する命令ビット数を圧縮すると共に、
- 20 当該プログラムの命令構造の中にレジスタ種類を記述して命令長の異なる圧縮プログラムが作成される。演算処理装置では、プログラム作成装置で作成された圧縮プログラムを取得してレジスタ種類を解読し、このレジスタ種類に基づいて当該レジスタを指定する命令ビット数を復元し、所定長さの命令に基づいて複数のレジスタを指定して任意の演算が実行される。従って、プログラム作成系ではレジスタの使用頻度に応じて命令の長さを可変できるので、頻繁にアクセスするレジスタに短い長さの命令をセットすることができる。これにより、プログラム実行系ではROM等に圧縮した命令をセットすることができ、プログラムデータを格納するROM等のメモリ容量を低減することができる。また、メモリセルや論理演算素子から成るPLDによりプロセッサを構築する場合に、RO
- 25

Mとして機能させるメモリセルの占有率を低減することができ、その分のメモリセルをレジスタに多く割り当てることができる。

本発明に係る第3の演算処理方法は、プログラム作成系で所定のプログラム言語に基づいてレジスタ相対レジスタアドレッシング演算を実行するための命令を編集してプログラムを作成し、プログラム実行系で当該プログラムと複数のレジスタとを使用してレジスタ相対レジスタアドレッシング演算を実行する演算処理方法である。プログラム作成系では、レジスタを使用する頻度に基づいて当該レジスタを指定する命令ビット数を圧縮すると共に、当該プログラムの命令構造の中にレジスタ種類を記述して命令長の異なる圧縮プログラムを作成し、プログラム実行系では、プログラム作成系で作成された圧縮プログラムを取得してレジスタ種類を解読し、ここで解読されたレジスタ種類に基づいて当該レジスタを指定する命令ビット数を復元し、ここで復元された所定長さの命令に基づいて複数のレジスタを指定して任意の演算を実行するようになされる。

この第3の演算処理方法によれば、プログラム作成系で所定のプログラム言語に基づいてレジスタ相対レジスタアドレッシング演算を実行するための命令を編集してプログラムを作成し、プログラム実行系で当該プログラムと複数のレジスタとを使用してレジスタ相対レジスタアドレッシング演算を実行する場合に、プログラム作成系ではレジスタの使用頻度に応じて命令の長さを可変できるので、頻繁にアクセスするレジスタに短い長さの命令をセットすることができる。従って、プログラム実行系ではROM等に圧縮した命令をセットすることができ、プログラムデータを格納するROM等のメモリ容量を低減することができる。また、メモリセルや論理演算素子から成るPLDによりプロセッサを構築する場合に、ROMとして機能させるメモリセルの占有率を低減することができ、その分のメモリセルをレジスタに多く割り当てることができる。

25

図面の簡単な説明

図1は、本発明に係る第1の実施例としての演算処理装置100の構成例を示すブロック図である。

図2は、レジスタアレイ11の内部構成例を示すブロック図である。

図3A～図3Eは演算処理装置100で取り扱う命令の構造例を示すフォーマット等である。図3Aはload、add及びcmp命令の構造例、図3Bはjump命令の構造例、図3Cは命令の構造例、図3Dはアクセス方法の内容例及び図3Eはフラグ状態の内容例である。

- 5 図4は、ROMセル14の演算プログラムによる演算命令の例を示す表図である。

図5は、レジスタr0～r12等の状態例を示す図である。

図6は、演算処理装置100の動作例を示すフローチャートである。

- 10 図7は本発明に係る第2の実施例としての演算処理装置100の構築例を示す半導体チップの平面図である。

図8は、演算処理装置100のレイアウト構築例を示す平面図である。

図9は、演算処理装置100の配線構築例を示す図である。

図10は、フラッシュメモリ70に格納される配線情報例を示す表図である。

図11は、演算処理装置100における結線例を示す図である。

- 15 図12は演算処理装置100の回路構築例を示す回路接続図である。

図13は、本発明に係る第3の実施例としての演算処理システム10の構成例を示すブロック図である。

図14は、演算処理システム10における処理例を示すフローチャートである。

- 20 図15は、本発明に係る第4の実施例としてのマイクロプロセッサ101の構成例を示すブロック図である。

図16は、レジスタアレイ11'の内部構成例を示すブロック図である。

- 図17A～図17Eはマイクロプロセッサ101で取り扱う命令の構造例を示すフォーマットである。図17Aは命令形態#F1に係るload、add、cmp命令の構造例、図17Bは命令形態#F2に係るload、add、cmp命令の構造例、図17Cは命令形態#F3に係るload、add、cmp命令の構造例、図17Dは命令形態#F4に係るload、add、cmp命令の構造例及び、図17Eは命令形態#F5に係るjump命令の構造例を各々示している。
- 25

図18A～図18Dは命令構造における記述内容例を示す表図である。図18

Aは命令の構造例、図18Bはアクセス方法#1、#2の内容例、図18Cはレジスタ種類1、2の内容例及び、図18Dはフラグ状態の内容例を各々示している。

図19は、プログラム作成系Iにおけるプログラム作成例を示す表図である。

- 5 図20は、プログラム作成装置200におけるコンパイル例を示すフローチャート（メインルーチン）である。

図21は、コンパイラにおける代入及び演算処理例を示すフローチャート（サブルーチン）である。

図22は、復元された演算プログラムによる演算命令の例を示す表図である。

- 10 図23は、r0、r1・・・r32、r33、r34、r35等のレジスタの状態例を示す図である。

図24は、外部メモリ2におけるデータ格納例を示す図である。

図25は、マイクロプロセッサ101における動作例を示すフローチャートである。

- 15 図26A及び図26Bは、命令ビット復元デコーダ13における処理例を示すフローチャートである。

発明を実施するための最良の形態

- 20 この発明は、従来の課題を解決したものであって、CPUや、RAM、ROM等の機能を1チップに集合化する場合に、レジスタの集合体をデータの随時書き込み及び読み出し可能なメモリのように動作できるようにすると共に、これらの機能部品等を個々に基板に配置する場合に比べて基板占有面積を低減できるようにした演算処理装置、その構築方法、演算処理システム及び演算処理方法を提供することを目的とする。

- 25 更にこの発明は、レジスタの使用頻度に応じて命令の長さを可変できるようにすると共に、頻繁にアクセスするレジスタには短い長さの命令をセットできるようにし、プログラムデータを格納するROM等のメモリ容量を低減できるようにした演算処理装置、その構築方法、演算処理システム及び演算処理方法を提供することを目的とする。

続いて、この発明に係る演算処理装置、その構築方法、演算処理システム及び演算処理方法の一実施の形態について、図面を参照しながら説明をする。

(1) 第1の実施例

図1は本発明に係る第1の実施例としての演算処理装置100の構成例を示す
5 ブロック図である。

この実施例では演算命令を実行するためのレジスタアレイ及び演算部を制御する命令実行制御部を備え、演算命令に基づいて一のレジスタを選択すると共に、このレジスタが保持する値によって他のレジスタを選択するレジスタ相対レジスタアドレッシング処理を実行して、レジスタアレイをデータの随時書込み及び読
10 出し可能なメモリ(RAM)のように動作できるようにすると共に、CPUや、RAM、ROM等を個々に基板に配置する場合に比べて基板占有面積を低減できるようにしたものである。

図1に示す演算処理装置100はプログラム可能な1チップマイクロコンピュータ等に適用して好適なプロセッサであり、任意の演算プログラムに基づいて演
15 算処理をするものである(第1の演算処理装置)。この例で演算処理装置100はプログラマブル・ロジック・デバイスを構成し、同一半導体チップ上に形成されたメモリセル及び算術論理演算素子を配線情報に基づいて結線し、複数のメモリセルによりレジスタアレイ11、及びROMセル14を構成し、複数の算術論理演算素子によりALU12、デコーダ13及び命令実行制御部50を構成した
20 ものである。演算処理装置100はインタフェース60を通じて外部のシステムバス等に接続される。

演算処理装置100はレジスタアレイ11を有している。レジスタアレイ11は複数のレジスタを有しており、各々のレジスタは書込みアドレスAw及び書込み制御信号Swに基づいて任意の値を保持し、及び、読出しアドレスArに基づ
25 いて被数Xや加数Y等の値を出力するようになされる。レジスタアレイ11には例えば、8192個×32bitのレジスタが設けられる。このレジスタアレイ11には演算部の一例となる算術論理演算ユニット(Arithmetic and Logic Unit: 以下でALUという)12がデータ信号線L20を通じて接続されており、レジスタアレイ11の中で指定されたレジスタから読み出されたXやY等の値を

演算するようになされる。演算結果の値はZである。演算種目は足し算、かけ算、引き算、わり算等である。演算種目は命令実行制御部50から出力されるALU制御信号S35に基づいて設定される。データ信号線L20には、ALU12の他にラッチ回路58、511、セクタ56、57、59等が接続されている。

- 5 データ信号線L20にはDATA、被数X値、加数Y値等が伝送される。

この演算処理装置100にはALU12を動作させるために命令解読部（以下でデコーダという）13が備えられている。デコーダ13には読出し専用メモリ（ROMセル）14が接続されており、ALU12の演算動作に必要な演算プログラムAPが機械語の命令（Instruction）で格納される。ROMセル14はプログラムカウンタ54からのカウント出力信号S5に基づいて演算プログラムAPを出力するようになされる。ROMセル14もプロセッサ内部に実装される。

デコーダ13では、予め準備された演算プログラムAPから演算命令を解読するようになされる。例えば、デコーダ13はROMセル14から読み出された機械語の命令を解読して命令制御信号S4、命令信号S9及び各引数信号S10を発生するようになされる。命令信号S9にはload命令、add命令、cmp命令、jump命令が含まれる。各引数信号S10にはアクセス方法#1、アクセス方法#2、レジスタ番号r0、r1・・・等、フラグ状態（flag condition）及びジャンプアドレス等が含まれる。命令制御信号S4は命令読出しステートマシーン52に出力される。この演算プログラムAPには、レジスタ相
20 対レジスタアドレッシング処理を実行するための演算命令を含んでいる。レジスタ相対レジスタアドレッシング処理とは演算命令に基づいて一のレジスタを選択し、ここで選択されたレジスタが保持する値によって他のレジスタを選択する処理をいう。この処理はアクセス方法#1によって実行される。

デコーダ13には命令実行制御部50が接続されており、デコーダ13によって解読された演算命令を実行するためにレジスタアレイ11及びALU12を制
25 御するようになされる。命令実行制御部50は実行ステートマシーン51、命令読出しステートマシーン52、セクタ53、プログラムカウンタ（PC）54、+1インクリメント55、第1のセクタ56、第2のセクタ57、入力用のセクタ59、ラッチ回路58、510、511を有しており、レジスタ相対レ

ジスタアドレッシング処理を実行するようになされる。命令読出しステートマシーン52ではデコーダ13から出力される命令制御信号S4に基づいてプログラムカウンタ54及び実行ステートマシーン51を制御する。例えば、当該マシーン52はデコーダ13から命令信号S9及び各引数信号S10が実行ステートマシーン51へ出力されると共に命令実行開始信号S29を出力する。

実行ステートマシーン51にはALU12、セクタ56、57、59、ラッチ回路58、510、511が接続されている。セクタ56の出力は読出しアドレス線L14を介在してレジスタアレイ11に接続され、その入力データ信号線L20及びアドレス信号線L33に各々接続されている。読出しアドレスAr又は読出し実行アドレスArのいずれかを選択するためである。セクタ57の出力は書込みアドレス線L13を介在してレジスタアレイ11に接続され、その入力データ信号線L20及びアドレス信号線L34に各々接続されている。書込みアドレスAw又は書込み実行アドレスAwのいずれかを選択するためである。当該マシーン51では命令実行開始信号S29に基づいて命令の実行を開始する。例えば、データの書込み時には、書込み制御信号Swがレジスタアレイ11に出力され、セクタ59には選択制御信号S24が出力される。データの読出し時には、選択制御信号S32がセクタ56に出力されると共に、読出しアドレスArが出力される。演算時には、ラッチ制御信号S34がラッチ回路58に出力され、ラッチ回路510にはラッチ制御信号S38が出力される。当該プロセッサ外部には外部制御信号S16が出力される。命令の実行が終了すると、実行ステートマシーン51は命令読み出しステートマシーン52へ実行終了信号S26を出力し、プログラムカウンタ54の値を進めるようになされる。

この実行ステートマシーン51及び命令読み出しステートマシーン52にはセクタ53が接続されており、選択制御信号S28に基づいてインクリメント出力信号S7又は分岐制御信号S27のいずれか一方を選択し、これをセクタ出力としてプログラムカウンタ54に出力するようになされる。選択制御信号S28は実行ステートマシーン51から供給される。インクリメント出力信号S7はインクリメンタ55からセクタ53へ出力される。プログラムカウンタ54ではカウント制御信号S30に基づいてROMセル14から演算プログラムAPを

読み出す場所が指定される。+1 インクリメンタ 5 5 はプログラムカウンタ 5 4 のカウント出力信号 S 5 を「+1」してインクリメンとするようになされる。カウント制御信号 S 3 0 は命令解読ステートマシーン 5 2 から供給される。

この例で実行ステートマシーン 5 1 に接続されたセクタ 5 6 では、一のレジスタを選択するための読出し実行アドレス A_r 又は当該レジスタを再度選択するための読出しアドレス A_r のいずれか一方を選択するようになされる。セクタ 5 6 は実行ステートマシーン 5 1 からの選択制御信号 S 3 2 によっていずれか一方のアドレスを選択するようになされる。読出し実行アドレス A_r は実行ステートマシーン 5 1 からセクタ 5 7 に出力される。読出しアドレス A_r はレジスタアレイ 1 1 から出力される。セクタ 5 6 の他に実行ステートマシーン 5 1 にはセクタ 5 7 が接続されており、一のレジスタを選択するための書込み実行アドレス A_w 又は当該レジスタを再度選択するための書込みアドレス A_w のいずれか一方を選択するようになされる。セクタ 5 7 は実行ステートマシーン 5 1 からの選択制御信号 S 3 1 によっていずれか一方のアドレスを選択するようになされる。書込み実行アドレス A_w は実行ステートマシーン 5 1 からセクタ 5 7 に出力される。書込みアドレス A_w はレジスタアレイ 1 1 から出力される。

セクタ 5 9 はデータバス 1 9 A、レジスタアレイ 1 1 及び A L U 1 2 に接続されており、データバス 1 9 A から取り込んだデータ (DATA)、レジスタアレイ 1 1 から出力される被数 X 値 (加数 Y 値) 又は A L U 1 2 から出力される演算結果値 Z のいずれかを選択制御信号 S 2 4 に基づいて入力制御するようになされる。ラッチ回路 5 8 はレジスタアレイ 1 1 の読出しポートと A L U 1 2 の間に接続されており、ラッチ制御信号 S 3 4 に基づいてレジスタ r_i の出力値 X をラッチするようになされる。ラッチ回路 5 1 0 は A L U 1 2 の比較出力部等に接続されており、ラッチ制御信号 S 3 8 に基づいて一致検出信号 S 2 2 をラッチして、フラグ状態 (flag condition) 信号 S 2 3 を出力するようになされる。ラッチ回路 5 1 1 はレジスタアレイ 1 1 の読出しポートとアドレスバス 1 9 B との間に接続されており、ラッチ制御信号 S 1 7 に基づいて外部アドレス (address) をラッチするようになされる。

なお、実行された命令によってジャンプ (命令分岐) が発生した場合は、ジャ

ンプ先のアドレスを示す分岐制御信号 S 2 7 を実行ステートマシン 5 1 からセレクト 5 3 へ出力される。セレクト 5 3 では選択制御信号 S 2 8 に基づいてその分岐制御信号 S 2 7 を選択し、この分岐制御信号 S 2 7 をプログラムカウンタ 5 4 へ書き込むようになされる。また、実行ステートマシーン 5 1 にはコントロール
5 バス 1 9 C が接続され、外部制御信号 S 1 6 を外部へ出力するようになされる。これは外部周辺機器を制御するためである。

図 2 はレジスタアレイ 1 1 の内部構成例を示すブロック図である。図 2 に示すレジスタアレイ 1 1 によれば、例えば、8 1 9 2 個の 3 2 b i t のレジスタ $r_0 \sim r_n$ ($i = 0 \sim n ; 8 1 9 1$) が備えられ、各々のレジスタ $r_0 \sim r_n$ の入力
10 には書き込みポート 1 5 が接続されている。1 b i t のレジスタは D 型のフリップ・フロップ回路等から構成される。書き込みポート 1 5 は図 1 に示したセクタ 5 9 に接続されており、書き込み制御信号 S w 及び書き込みアドレス A w に基づいて、データバス 1 9 A から取り込んだデータ (DATA)、レジスタアレイ 1 1 から出力される被数 X 値 (加数 Y 値) 又は A L U 1 2 から出力される演算結果値 Z のいずれか
15 をレジスタ $r_0 \sim r_n$ に書き込むようになされる。書き込みポート 1 5 は書き込みアドレス線 L 1 3 を介在して図 1 に示したセクタ 5 7 に接続されている。書き込み実行アドレス A w 又は書き込みアドレス A w を供給するためである。

各々のレジスタ $r_0 \sim r_n$ の出力には読出しポート 1 6 が接続されている。読出しポート 1 6 はデータ信号線 L 2 0 を通じて図 1 に示した A L U 1 2、ラッチ
20 回路 5 8、5 1 1、セクタ 5 6、5 7、5 9 等に接続されており、読出しアドレス A r に基づいて指定されたレジスタ r_i からデータ (DATA) を読み出すようになされる。読出しポート 1 6 は読出しアドレス線 L 1 4 を介在して図 1 に示したセクタ 5 6 に接続されている。読出し実行アドレス A r 又は読出しアドレス A r を供給するためである。書き込みアドレス線 L 1 3 を介在して図 1 に示したセ
25 レクタ 5 7 に接続されている。書き込み実行アドレス A w 又は書き込みアドレス A w を供給するためである。

次に、演算処理装置 1 0 0 で取り扱う命令の構造例を示すフォーマット等について説明をする。図 3 A に示すフォーマットによれば、命令は 3 2 ビットの固定長であり、大きく分けて l o a d 命令や、a d d 命令、c m p 命令等と j u m p

命令とによって取り扱われる。load、add及びcmp命令においては、最初の2ビットが命令の種類を表している。命令は図3Cに示すように4種類である。命令は図3Cにおいて、コード「0」でloadを示し転送命令、コード「1」でaddを示し加算命令、コード「2」でcmpを示し比較命令である。

- 5 コード「3」はjumpで分岐命令を示している。cmp命令に関しては比較結果が同じであった場合は、図1に示したラッチ回路510のフラグ状態信号S23に基づいてzero flagが1にセットされ、同じでなかった場合は0がセットされる。

- 図3Aに示す命令に続く2ビットにはアクセス方法#1が記述され、それに続く2bitにはアクセス方法#2が記述される。Operandは左がアクセス方法#1、レジスタ番号1で表され、右がアクセス方法#2、レジスタ番号2で表される。つまり、アクセス方法#1はレジスタ番号1で示されるレジスタriのアクセス方法を示しており、アクセス方法#2はレジスタ番号2で示されるレジスタriのアクセス方法を示している。アクセス方法#1とアクセス方法#2はそれぞれ
15 レジスタ番号No. 1、レジスタ番号No. 2に対応し、これらの間で処理が行われる。

- いずれも、図3Dに示すようにアクセス方法は4種類が準備されている。図3Dにおいて、コード「0」は「レジスタ直接」でレジスタ番号で示されるレジスタriの値を直接用いることを示している。コード「1」は「レジスタ相対レジスタ」でレジスタ番号で示されるレジスタの値を再度レジスタ番号と解釈し、そのレジスタ番号で示されるレジスタの値を用いることを示している。コード
20 「2」は「レジスタ相対外部」でレジスタ番号で示されるレジスタ値を外部アドレスとして扱い、当該演算処理装置100の外部に対してアクセスを示している。コード「3」は未使用である。アクセス方法#1とアクセス方法#

- 25 2はそれぞれレジスタ番号1、レジスタ番号2に対応し、これらの間で処理が行われる。例えば、レジスタ番号1は被数を保持するレジスタriを示し、レジスタ番号2は加数を保持するレジスタriを示す。

また、図3Bに示すjump命令のフォーマットによれば、最初の2bitに命令が記述され、続く2bitにはフラグ状態(flag condition)が記述される。

続く 28 bit にはジャンプアドレスが記述される。フラグ状態は図 3 E に示すように、命令実行制御を移すかどうかの判断をするための条件である。コード「0」は「無条件」で常に制御を移す。コード「1」は「zero flag」で zero flag が「1」である場合に、制御を移す。コード「2」は「non-zero flag」で zero flag が「0」である場合に制御を移すようになされる。コード「3」は未使用である。

図 4 は ROM セル 14 の演算プログラム AP による演算命令の例を示す表図である。図 5 はレジスタ r 0 ~ r 12 等の状態例を示す図である。この例ではレジスタアレイ 11 の 13 個のレジスタ r i (i = 0 ~ 12) のいずれかを使用して、そのレジスタ r i に格納された値に「1」を加算し、そのレジスタ r i の値が示す任意のレジスタに演算結果を格納する場合を想定する。図 4 に示す演算命令 (Instruction) # I 1 ~ # I 4 は ROM セル 14 の演算プログラム AP に記述されたものである。

各々の演算命令 # I 1 ~ # I 4 には、ニーモニックによる表現、機械語による表現及び処理の内容が示されている。図 4 に示す演算命令 # I 1 は図 3 A に示した命令構造において、機械語で 5 0 0 1 4 0 0 B h によって表される add [r 1 0] , r 1 1 であり、レジスタアレイ 11 の r 1 0 で示されるレジスタ番号のレジスタ r 1 0 の値に、 r 1 0 で示されるレジスタ番号のレジスタ r 1 1 の値を加算し、その結果を r 1 0 で示されるレジスタ番号のレジスタ r 1 0 に格納する内容である。

演算命令 # I 2 は機械語で 4 0 0 1 4 0 0 B h によって表される add r 1 0 , r 1 1 であり、レジスタ r 1 0 の値にレジスタ r 1 1 の値を加算してその演算結果をレジスタ r 1 0 に書き戻す内容である。演算命令 # I 3 は機械語で 8 0 0 1 4 0 0 C h によって表される cmp r 1 0 , r 1 2 であり、レジスタ r 1 0 の内容とレジスタ r 1 2 の内容とを比較し、同じ値の場合は zero flag に「1」をセットし、異なっている場合は「0」にセットする内容である。命令 # I 4 は機械語で E 0 0 0 0 0 0 0 h によって表される jump nz , LOOP であり、zero flag が「0」の場合は、LOOP で示されるラベルへ制御を移す内容である。

図 5 に示すレジスタアレイ 11 の 13 個のレジスタ状態例によれば、レジスタ

r 0 ~ r 1 0 の初期値はいずれも「0」であり、レジスタ r 1 1 の初期値は「1」、レジスタ r 1 2 の初期値は「1 0」である。図 4 に示した命令 # I 1 を実行する場合、図 5 に示すレジスタ r 1 0 の値は「0」であるので、レジスタ r 0 が選択される。そのレジスタ r 0 の値である「0」が読出しポート 1 6 に読み出され、続いてレジスタ r 1 1 の値である「1」が読出しポート 1 6 に読み出され、これらの値が加算され、その演算結果である「1」がレジスタ r 0 に格納される。これにより、レジスタ r 0 の値が「1」ずつ増加するようになされる。この例ではレジスタ r 1 2 の値である 1 0 回まで加算処理が繰り返される。

図 4 に示した命令 # I 2 を実行する場合は、図 5 に示すレジスタ r 1 0 の値である「0」にレジスタ r 1 1 の値である「1」が加算され、その演算結果の「1」がレジスタ r 1 0 に格納される。これにより、次のレジスタへ処理が移るようになされる。図 4 に示した命令 # I 3 を実行する場合、zero flag の値がラッチ回路 5 1 0 によって保持され、以降の命令によって参照される。これを前提にしてレジスタ r 1 0 の値「0」とレジスタ r 1 2 の値である「1 0」とが比較される。

この例では、レジスタ r 1 0 の値「0」とレジスタ r 1 2 の値「1 0」とが異なるので、zero flag には「0」がセットされる。zero flag に「1」がセットされるのは、レジスタ r 1 0 の値「1 0」とレジスタ r 1 2 の値「1 0」とが一致した場合である。図 4 に示した命令 # I 4 を実行する場合、zero flag が「0」なので制御を命令 # I 1 に移す (LOOP)。上記の動作が 1 0 回、繰り返されるとレジスタ r 1 0 の値が「1 0」になり、レジスタ r 1 0 の値「1 0」とレジスタ r 1 2 の値「1 0」とが一致するので、命令 # I 3 により zero flag が「1」にセットされ、命令 # I 4 で制御が命令 # I 1 に移らなくなり、演算処理が終了する。

続いて、本発明に係る第 1 の演算処理方法について当該演算処理装置 1 0 0 の動作例を説明する。図 6 は演算処理装置 1 0 0 の動作例を示すフローチャートである。

この実施例では、図 4 に示した演算命令 # I 1 ~ # I 4 を形成する演算プログラム AP に基づいて加算処理を実行する場合を想定する。もちろん、演算処理装置 1 0 0 には書込みアドレス Aw 及び書込み制御信号 Sw に基づいて任意の値を

- 保持し、及び、読出しアドレスA_rに基づいて当該値を出力するレジスタアレイ11が予め準備されている場合を前提とする。レジスタアレイ11のレジスタ状態については、図5に示したように、例えば、13個のレジスタr₀～r₁₀の初期値がいずれも「0」であり、レジスタr₁₁の初期値が「1」、レジスタr₁₂の初期値が「10」である。これらの初期値を書き込む場合は、実行ステートマシーン51ではアドレス信号線L₃₄に書き込みアドレスA_wが出力され、セクタ57が選択制御信号S₃₁に基づいてレジスタr₁₀、レジスタr₁₁及びレジスタr₁₂を選択することで、その初期値「0」、「1」、「10」が設定される。
- 10 これを動作条件にして、図6に示すフローチャートのステップA1で、まず、デコーダ13はROMセル14から演算プログラム（機械語命令）APを受け取り、演算プログラムAPを解読して演算命令#I₁～#I₄を検出する。このとき、デコーダ13では図4で説明した機械語の5001400B_hによって示されるadd [r₁₀]、r₁₁に係る演算命令#I₁や、機械語で4001400B_hによって表されるadd r₁₀、r₁₁に係る演算命令#I₂、機械語で8001400C_hによって表されるcmp r₁₀、r₁₂に係る演算命令#I₃、機械語でE0000000_hによって表されるjump nz, LOOPに係る演算命令#I₄を検出し、これらの演算命令#I₁～#I₄から、命令制御信号S₄、命令信号S₉及び各引数信号S₁₀が生成され、これらの信号S₉及びS₁₀が実行ステートマシーン51に出力するようになされる。命令信号S₉にはload命令、add命令、cmp命令、jump命令が含まれる。各引数信号S₁₀にはアクセス方法#1、アクセス方法#2、レジスタ番号r₀、r₁・・・等、フラグ状態（flag condition）及びジャンプアドレス等が含まれる。命令制御信号S₄はデコーダ13から命令読出しステートマシーン52に出力される。
- 25

その後、ステップA2で実行ステートマシーン51は命令読出しステートマシーン52の命令読出し制御を受けて演算命令#I₁を受け取り、5001400B_hによって示されるadd [r₁₀]、r₁₁に基づいて一のレジスタr₁₀を選択する。このとき、アドレス信号線L₃₃にレジスタの番号「10」を出

力する。選択制御信号S 3 2を用いてセクタ5 6がアドレス信号線L 3 3を選択するようにする。これにより、読出しアドレス線L 1 4にレジスタ番号「1 0」が出力される。この値「1 0」はレジスタアレイ1 1でレジスタ番号1 0のレジスタr 1 0を読み出すアドレスA_rとして用いられる。レジスタアレイ1 1
5 5ではレジスタr 1 0が選択される。レジスタr 1 0の値は「0」である。

そして、ステップA 3でレジスタr 1 0が保持する値「0」によってレジスタ
相対レジスタアドレッシング処理を実行する。このとき、レジスタr 1 0の値
「0」は読出しポート1 6からデータ信号線L 2 0に出力するようになされる。
他のレジスタr 0を選択するためである。これにより、レジスタアレイ1 1から
10 データ信号線L 2 0に出力されたレジスタr 0の値「0」はラッチ回路5 8で保
持される。これと共に、実行ステートマシーン5 1は選択制御信号S 3 2に基づ
いてデータ信号線L 2 0を選択するようになされる。これにより、セクタ5 6は
読出しアドレス線L 1 4にレジスタr 1 0の値「0」を出力するようになる。
レジスタアレイ1 1ではレジスタr 0が選択される。

15 そして、ステップA 4に移行して演算命令# I 1に基づいて他のレジスタr 1
1を選択する。このとき、実行ステートマシーン5 1は選択制御信号S 3 2をセ
クタ5 6に出力してセクタ5 6により読出し実行アドレスA_rを選択するよ
うにする。これ共に、アドレス信号線L 3 3に読出し実行アドレスA_rとして値
「1 1」を出力する。この値「1 1」はレジスタアレイ1 1でレジスタ番号「1
20 1」のレジスタr 1 1を読み出すアドレスとして用いられる。レジスタr 1 1の
値は「1」である。これにより、読出しアドレスA_rに基づいてレジスタr 1 1
が選択される。この選択によって読出しポート1 6からデータ信号線L 2 0へレ
ジスタr 1 1の出力値であるY値＝「1」を出力するようになされる。これらの
操作により、データ信号線L 2 0に出力されたレジスタr 1 1のY値＝「1」と、
25 ラッチ回路5 8の出力であるX値＝「0」とがALU 1 2に入力される。つまり、
レジスタr 1 0が保持する値「0」によって指定されたレジスタr 0のX値＝
「0」（レジスタ相対レジスタアドレッシング処理によって選択されたレジスタ
r 0の値）とレジスタr 1 1のY値＝「1」とがALU 1 2に入力される。

そして、ステップA 5でレジスタr 1 1が保持するY値＝「1」とレジスタ相

対レジスタアドレッシング処理によって選択されたレジスタ r_0 の X 値 = 「0」
とが ALU12 によって加算するようになされる。このとき、ALU12 には加
算命令を示す演算命令信号 S35 が入力されている。これにより、ALU12 で
は X = 「0」 値に Y 値 = 「1」 を加算し、その加算結果値 Z = 「1」 を演算結果
5 信号 S21 としてセクタ 59 に出力するようになされる。

そして、ステップ A6 で機械語で 4001400Bh によって表される演算命
令 #I2 に基づいて演算結果値 Z がセクタ 59 によって選択される。このとき、
セクタ 59 では選択制御信号 S24 により演算結果値 Z を選択するようにセッ
トされる。演算結果値 Z がレジスタアレイ 11 に入力される。レジスタアレイ 1
10 1 ではレジスタ相対レジスタアドレッシング処理によって指定されるレジスタ r_0
に、書込みアドレス Aw 及び書込み制御信号 Sw に基づいて演算結果値 Z を格
納するようになされる。つまり、レジスタ r_{10} が指定するレジスタ r_0 に AL
U12 の演算結果値 Z を書き戻す場合は、まず、実行ステートマシーン 51 はア
ドレス信号線 L33 には書込み実行アドレス Aw として値 「10」 を出力する。
15 このとき、読出し用のセクタ 56 には選択制御信号 S32 を出力してアドレス
信号線 L33 を選択するようになされる。

そして、レジスタアレイ 11 の読出しポート 16 にはレジスタ r_{10} の値
「0」 が出力されるので、今度は、書込み用のセクタ 57 には選択制御信号 S
31 を出力してデータ信号線 20 を選択するようになされる。この選択によって
20 ジスタアレイ 11 の読出しポート 16 からセクタ 57 を通じてレジスタ r_0 を
選択する値 「0」 が書込みアドレス Aw としてレジスタアレイ 11 に入力され
る。ここまでのレジスタアレイ 11 には書込みアドレス Aw と書き込む内容が
入力される。その後、実行ステートマシーン 51 は書込み信号 SW を用いて実
際に値の書き込みを指示する。このようにして、演算命令 #I1 の add [r
25 10], r11 を演算することができる。

この例では、ステップ A7 でレジスタ r_{12} が示す値 = 10 回に至ったかが判
別される。この際の判別では演算命令 #I3 に基づいてラッチ回路 510 に保持
されている zero flag の値が参照される。zero flag の値はレジスタ r_{10} の値と
レジスタ r_{12} の値 「10」 とが比較され、両方の値が異なる場合は zero flag に

「0」がセットされる。両方の値が一致する場合はzero flagに「1」がセットされる。従って、レジスタr12が示す値=10回に至っていない場合はzero flagに「0」がセットされることから、演算命令#I4に基づいてステップA5に戻って（LOOPして）レジスタr11の値である「1」が読出しポート16に読み出される。この値「1」が繰り返し加算処理がなされ、レジスタr0の値が「1」ずつ増加するようになされ、その演算結果である「1」、「2」、「3」・・・がレジスタr0に格納される。これにより、書込みアドレスAw及び書込み制御信号Swに基づいて演算結果値Zをレジスタr0に格納することができる。そして、ステップA7でレジスタr12が示す値=10回に至った場合は、レジスタr10の値「10」とレジスタr12の値「10」とが一致するので、命令#I3によりzero flagが「1」にセットされ、命令#I4で制御が命令#I1に移行されなくなるので、ステップA5にLOOPすることなく演算処理が終了する。

このように、本発明に係る第1の実施例としての演算処理装置及び演算処理方法によれば、図6に示したフローチャートのステップA1でデコーダ13によって解読された演算命令#I1～#I4等は命令信号S9及び各引数信号S10となって実行ステートマシーン51へ出力される。実行ステートマシーン51ではステップA2で演算命令#I1～#I4に基づいて一のレジスタr10を選択し、ステップA3でレジスタr10が保持する値「0」によってレジスタ相対レジスタアドレッシング処理を実行するようになされる。また、ステップA4で他のレジスタr0が選択されると、ステップA5でX値とY値とがALU12によって加算され、レジスタアレイ11のレジスタr0ではレジスタ相対レジスタアドレッシング処理によって指定される書込みアドレスAw及び書込み制御信号Swに基づいて演算結果値Zを保持し、及び、ステップA5で読出しアドレスArに基づいて当該演算結果値Zを次の演算の加数X値として出力するようになされる。ALU12ではレジスタアレイ11から読み出されたX値及びY値を加算するようになされる。

従って、レジスタアレイ11をデータの随時書込み及び読出し可能なメモリ（RAM）のように取り扱うことができるので、レジスタr0～r12等を通常

のプロセッサのRAM（メモリ）のように扱いながら、レジスタアレイ11（レジスタ）にアクセスをすることが可能となる。これにより、従来方式の中央演算処理装置（CPU）に比べて外部メモリの格納番地を指定するレジスタ相対メモリアドレッシング処理に依存しなくなる。従来方式のCPUの中にRAMやROMの機能を取り込むことができるので、高速演算処理を実行すること、及び、CPUや、RAM、ROM等を個々に基板に配置する場合に比べて基板占有面積を低減することができる。当該演算処理装置100が適用される応用装置のコンパクト化を図ることができる。

（２）第２の実施例

図7は本発明に係る第２の実施例としての演算処理装置100の構築例を示す半導体チップの平面図である。

この実施例では図1に示した演算処理装置100を構築する場合に、予め同一半導体チップ上に複数のメモリセル及び算術論理演算素子を形成し、このメモリセルを組み合わせるレジスタアレイ11及びROMセル14を画定すると共に、算術論理演算素子を組み合わせるALU12、デコーダ13及び命令実行制御部50を画定する。その後、レジスタアレイ11、ROMセル24、ALU12、デコーダ13及び命令実行制御部50を予め設定された配線情報に基づいて結線すると共に、ROMセル14に上述した演算プログラムAPを書込むようになされる。

図7に示す演算処理装置100は予め同一半導体チップ上に複数のメモリセルMSE及び算術論理演算素子LAYを形成したプログラム可能な半導体デバイス（PLD；Programmable Logic Device）である。この半導体デバイスの行方向には例えば $N=7$ 本（ $i=1, 2, \dots, N$ ）の配線パターン（以下で行配線CO1～CO7という）が配置される。列方向には同様に $M=7$ 本（ $j=1, 2, \dots, M$ ）の配線パターン（以下で列配線RO1～RO7という）が配置される。この行配線CO1～CO7と直交するように配置された列配線RO1～RO7によってPLDが格子状を構成するようになされている。各々の格子内には複数のメモリセルMSE及び算術論理演算素子LAYを一単位とするメモリセル／算術論理演算素子ブロック（以下で単にセルブロックという）SE i

j (i = 1 ~ 8、j = 1 ~ 8) が設けられている。

この例では $8 \times 8 = 64$ 個のセルブロック SE 1 1 ~ SE 8 8 が同一半導体チップ上に形成されている。1つのセルブロック SE i j の上半分には複数のメモリセル MSE が配置され、その下半分には複数の算術論理演算素子 LAY が配置
5 されている。また、レジスタアレイ 1 1 及び ROM セル 1 4 はセルブロック SE i j 内のメモリセルを組み合わせる場合と、他のセルブロック SE i j 内のメモリセルをも組み合わせる場合とを想定している。

各々の行配線 CO 1 ~ CO 7 の両終端側及び列配線 RO 1 ~ RO 7 の両終端側
10 には I/O インタフェース 6 0 ~ 6 3 が設けられ、I/O インタフェース 6 1 には行配線 CO 1 ~ CO 7 の一端側が接続され、その他端側には I/O インタフェース 6 3 が接続されている。I/O インタフェース 6 0 には列配線 RO 1 ~ RO 7 の一端側が接続され、その他端側には I/O インタフェース 6 2 が接続されている。セルブロック SE 1 1 に最も近い行配線 CO 1 や、列配線 RO 1 等を使用
15 して I/O インタフェース 6 0 又は I/O インタフェース 6 3 に配線をプログラムすることが可能となされている。もちろん、他の行配線 CO 2 ~ CO 7 や、列配線 RO 2 ~ RO 8 等を使用して他のセルブロック SE i j 1 に配線をプログラムすることが可能となされている。

4つの I/O インタフェース 6 0 ~ 6 3 は例えば、格子内の 64 個のセルブ
20 ック SE i j の内、外周縁部を構成するセルブロック SE 1 1, SE 1 2 . . . SE 1 8、SE 1 1, SE 2 1 . . . SE 8 1、SE 8 1、SE 8 2 . . . SE 8 8、及び SE 1 8, SE 2 8 . . . SE 8 8 に沿ってこれらを取り囲むように配置されている。各セルブロック SE i j 内のメモリセル間、算術論理演算素子間の配線や、セルブロック SE 1 1 ~ SE 8 8 間の配線、これらの間を結線する
25 列配線 RO 1 ~ RO 7 と I/O インタフェース間には複数のスイッチ素子（トランジスタ）が至る所に配置され、配線情報に基づいてスイッチ素子を ON することで、回路要素間を自在に接続できるようになされている。

なお、この段階ではレジスタアレイ 1 1、ALU 1 2、デコーダ 1 3、ROM セル 1 4 及び命令実行制御部 5 0 等が画定されておらず、汎用性を有したプログ

ラマブル・ロジック・デバイスを成している。この例では、プログラマブル・ロジック・デバイスを電子機器にCPUとして組み込む場合に、各機能回路にセルブロックSE11～SE88を割り当てたり、これらの間を列配線RO1～RO7や、行配線CO1～CO7等を使用して結線し、当該電子機器で要求される演算機能に対して自在にプログラムしてユーザの希望に合ったCPU機能等を構築するようになされる。

続いて、演算処理装置100のレイアウト構築例について説明する。図8においては、図7で準備された半導体チップ上に、図8に示す波線で囲むようなレジスタアレイ11、ALU12、デコーダ13、ROMセル14、実行ステートマシーン51、命令読出しステートマシーン52、セクタ53、プログラムカウンタ54、インクリメンタ55、セクタ56、セクタ57、入力用のセクタ59、ラッチ回路58、510、511を画定して演算処理装置100をレイアウトする。

なお、実行ステートマシーン51、命令読出しステートマシーン52、セクタ53、プログラムカウンタ54、インクリメンタ55、セクタ56、セクタ57、入力用のセクタ59及びラッチ回路58、510、511は図1に示した命令実行制御部50を構成する。この例で、レジスタアレイ11には例えば、セルブロックSE1、SE14、SE23及びSE43を割り当てて構成する。8192個のレジスタ $r_0 \sim r_n$ にはこれらのセルブロックSE1、SE14、SE23及びSE43のメモリセルMSEを使用する。レジスタアレイ11は書込みアドレスAw及び書込み制御信号Swに基づいて任意の値を保持し、及び、読出しアドレスArに基づいて当該値を出力する複数のレジスタ $r_0 \sim r_n$ を備えるようにメモリセルMSEを割り当てる。

図2に示した書込みポート15及び読出しポート16には、例えば、セルブロックSE1、SE14、SE23及びSE43の算術論理演算素子LAYの入力バッファ回路や、出力バッファ回路を使用するとよい。これらの間の配線には列配線RO1、RO2及び行配線CO2、CO3、CO4を使用して結線するようになされる。ALU12にはセルブロックSE26及びSE36等を割り当てて構成する。セルブロックSE26及びSE36の算術論理演算素子LAYを使用

してALU12を構築する。これらの間の配線には列配線RO1、RO2及び行配線CO5、CO6等を使用して結線するようになされる。

- デコーダ13にはセルブロックSE54及びSE64等を割り当てて構成する。セルブロックSE54及びSE64等の算術論理演算素子LAYを使用してデコーダ13を構築する。これらの間の配線には列配線RO5及び行配線CO3、CO4等を使用して結線するようになされる。ROMセル14にはセルブロックSE52及びSE62等を割り当てて構成する。セルブロックSE52及びSE62等のメモリセルMSEを使用して読出し専用メモリを構築する。算術論理演算素子LAYの入力バッファ回路や、出力バッファ回路を使用して演算プログラムAPの書込み／読出し回路を構成するようになされる。これらの間の配線には列配線RO5及び行配線CO1、CO2等を使用して結線するようになされる。

- 実行ステートマシーン51にはセルブロックSE56、SE57、SE58、SE66、SE67及びSE68等を割り当てて構成する。これらの間の配線には列配線RO5及び行配線CO5、CO6、CO7等を使用して結線するようになされる。実行ステートマシーン51は演算命令#I1～#I4に基づいて一のレジスタriを選択し、ここで選択されたレジスタriが保持する値によって他のレジスタを選択するレジスタ相対レジスタアドレッシング処理を実行するものである。命令読出しステートマシーン52にはセルブロックSE85及びSE86等を割り当てて構成する。これらの間の配線には列配線RO7及び行配線CO4、CO5、CO6等を使用して結線するようになされる。セクタ53及びプログラムカウンタ54には、セルブロックSE73等を割り当てて構成する。他の回路との間の配線には列配線RO7及び行配線CO2等を使用して結線するようになされる。インクリメンタ55にはセルブロックSE83等を割り当てて構成する。他の回路との間の配線には列配線RO7及び行配線CO2、CO3等を使用して結線するようになされる。

セクタ56にはセルブロックSE33等を割り当てて構成する。他の回路との間の配線には行配線CO3及びCO4等を使用して結線するようになされる。セクタ56は一のレジスタを選択するための読出し実行アドレスAr又は当該レジスタを再度選択するための読出しアドレスArのいずれか一方を選択するも

のである。セクタ 57 にはセルブロック SE 34 等を割り当てて構成する。他の回路との間の配線には行配線 CO 2 及び CO 3 等を使用して結線するようになされる。セクタ 57 は一のレジスタ r_i を選択するための書込み実行アドレス又は当該レジスタを再度選択するための書込みアドレス A_w のいずれか一方を選択するものである。

入力用のセクタ 59 にはセルブロック SE 12、SE 22 等を割り当てて構成する。他の回路との間の配線には列配線 RO 1 及び行配線 CO 1 等を使用して結線するようになされる。ラッチ回路 58 にはセルブロック SE 35 等を割り当てて構成する。他の回路との間の配線には列配線 RO 2 及び行配線 CO 5 等を使用して結線するようになされる。ラッチ回路 510 にはセルブロック SE 37 等を割り当てて構成する。他の回路との間の配線には列配線 RO 2 及び行配線 CO 6、CO 7 等を使用して結線するようになされる。ラッチ回路 511 にはセルブロック SE 41 等を割り当てて構成する。他の回路との間の配線には列配線 RO 3 及び行配線 CO 1 等を使用して結線するようになされる。

次に、演算処理装置 100 の配線構築例について説明をする。図 9 においては、図 8 で画定されたレジスタアレイ 11、ALU 12、デコーダ 13、ROM セル 14、実行ステートマシーン 51、命令読出しステートマシーン 52、セクタ 53、プログラムカウンタ 54、インクリメンタ 55、セクタ 56、セクタ 57、入力用のセクタ 59、ラッチ回路 58、510、511 間を配線情報により結線して演算処理装置 100 を構築するようになされる。この例では電子機器の例えば、プリント配線基板に、演算処理装置 100 を構築するプログラマブル・ロジック・デバイスを取り付けると共に、書き換え可能な不揮発性の記憶装置の一例となるフラッシュメモリ 70 が取り付けられる。

このフラッシュメモリ 70 には、各セルブロック SE ij 内のメモリセル間、算術論理演算素子間の配線や、セルブロック SE 11～SE 88 間の配線、これらの間を結線する列配線 RO 1～RO 7 と I/O インタフェース間等の至る所に配置されたスイッチ素子を ON するための配線情報が格納されている。

図 10 はフラッシュメモリ 70 に格納される配線情報例を示す表図である。図 10 に示す配線情報例では配線情報 D 1～D 38・・・に対する信号線 L 1～L

38・・・と共に、当該信号線L1～L38に伝送される信号又はDATA等の用途を示している。配線情報D1～D38・・・は回路技術言語により作成され、ネットリスト（結線情報）として準備される。この例でフラッシュメモリ70に格納される配線情報D1～D38・・・は当該演算処理装置100の機能に応じて

5 随時書き換えられるものである。

なお、演算プログラムAPはROMセル14に書込むようになされる。この例で演算プログラムAPには、レジスタ相対レジスタアドレッシング処理を実行するための演算命令#I1～#I4が含まれている。この演算プログラムAPを配線情報D1～D38・・・とリンクさせてデコーダ13に出力するようにしても
10 よい。フラッシュメモリ70によってROMセル14の機能を兼用できるようになる。

次に、演算処理装置100における結線例について説明をする。図11に示す結線例によれば、例えば、セルブロックSE22と、セルブロックSE66とを信号線L24で接続する場合、セルブロックSE22と列配線R03との間に配置されたスイッチング用のトランジスタT23と、列配線R03と行配線C06との間に配置されたトランジスタT63と、行配線C06とセルブロックSE66との間に配置されたトランジスタT66とを使用する。各々のトランジスタT23、T63、T66はゲート制御可能になされている。これを前提にして、電源オンと共にフラッシュメモリ70から、例えば図11に示した表図の配線情報
15 D24が読み出される。配線情報D24はトランジスタT23、T63、T66の各々のゲートに設定される。これにより、各々のトランジスタT23、T63、T66がONする。これらのトランジスタT23、T63、T66のONにより、セルブロックSE22と列配線R03との間、列配線R03と行配線C06との間及び行配線C06とセルブロックSE66との間が電氣的に接続され、信号線
20 L24を構築することができる。この例でセルブロックSE22はセクタ59に割り当てられ、セルブロックSE66は実行ステートマシーン51に割り当てられており、信号線L24は当該セクタ59の選択制御信号S24を伝送する際に使用される。

このように、各セルブロックSEij内のメモリセル間等の至る所に配置され

たスイッチング用のトランジスタに配線情報D 1 ~ D 3 8 . . . 等が設定され、この配線情報D 1 ~ D 3 8 . . . 等に基づいて当該トランジスタにより回路要素間を自在に接続する。これにより、レジスタアレイ 1 1、ALU 1 2、デコーダ 1 3、ROMセル 1 4 及び命令実行制御部 5 0 等を結線するようになされる。この設定によりROMセル 1 4 に格納される演算プログラムAPのサイズを小さくすることができる。

続いて、演算処理装置 1 0 0 の回路構築例について説明をする。図 1 2 は電源オンと共に結線された演算処理装置 1 0 0 の回路図を示している。図 1 に示した演算処理装置 1 0 0 の構成図に関して複数の信号線等を記述したものである。ここで記述した信号線はフラッシュメモリ 7 0 に記録される配線情報D 1 ~ D 3 8 等に基づくものである。配線情報D 1 ~ D 3 8 等は信号線毎に準備される。この例では配線情報D 3 に基づいて図 1 1 に示したようなスイッチング用のトランジスタがONすることで、信号線L 3 がデコーダ 1 3 とROMセル 1 4 を接続する。信号線L 3 には演算プログラムAPが伝送される。同様にして信号線L 4 は配線情報D 4 に基づいてデコーダ 1 3 と命令読出しステートマシーン 5 2 とを接続する。信号線L 4 には命令制御信号S 4 が伝送される。

信号線L 5 は配線情報D 5 に基づいてROMセル 1 4 と命令読出しステートマシーン 5 2 とプログラムカウンタ 5 4 とを接続する。信号線L 5 にはカウント出力信号S 5 が伝送される。信号線L 6 は配線情報D 6 に基づいてセクタ 5 3 とプログラムカウンタ 5 4 とを接続する。信号線L 6 にはセクタ出力として分岐制御信号S 2 7 又はインクリメント出力信号S 7 のいずれかが伝送される。信号線L 7 は配線情報D 7 に基づいてインクリメンタ 5 5 とセクタ 5 3 とを接続する。信号線L 7 にはインクリメント出力信号S 7 が伝送される。信号線L 9、信号線L 1 0 は配線情報D 1 0 に基づいてデコーダ 1 3 と実行ステートマシーン 5 1 とを接続する。信号線L 9 には命令信号S 9 が伝送され、信号線L 1 0 には各引数信号S 1 0 が伝送される。信号線L 1 1 は配線情報D 1 1 に基づいてデータバス 1 9 A と入力用のセクタ 5 9 とを接続する。信号線L 1 1 にはデータ (DATA) が伝送される。信号線L 1 2 は配線情報D 1 2 に基づいてレジスタアレイ 1 1 と実行ステートマシーン 5 1 とを接続する。信号線L 1 2 には書込み制御信号

S_wが伝送される。

書込みアドレス線L 1 3は配線情報D 1 3に基づいてレジスタアレイ1 1とセ
レクタ5 7とを接続する。書込みアドレス線L 1 3には書込みアドレスA_wが伝
送される。読出しアドレス線L 1 4は配線情報D 1 4に基づいてレジスタアレイ
5 1 1とセクタ5 6とを接続する。読出しアドレス線L 1 4には読出しアドレス
A_rが伝送される。信号線L 1 5は配線情報D 1 5に基づいてラッチ回路5 1 1
とアドレスバス1 9 Bとを接続する。信号線L 1 5には外部アドレスが伝送され
る。信号線L 1 6は配線情報D 1 6に基づいて実行ステートマシーン5 1とコン
トロールバス1 9 Cとを接続する。信号線L 1 6には外部制御信号S 1 6が伝送
10 される。信号線L 1 7は配線情報D 1 7に基づいてラッチ回路5 1 1と実行ステ
ートマシーン5 1とを接続する。信号線L 1 7にはラッチ制御信号S 1 7が伝送
される。

データ信号線L 2 0は配線情報D 2 0に基づいてレジスタアレイ1 1と、AL
U 1 2と、データバス1 9 Aと、セクタ5 7と、セクタ5 6と、5 8と、入
15 力用のセクタ5 9とを接続する。データ信号線L 2 0にはデータ (DATA)、X
値、Y値等が伝送される。信号線L 2 1は配線情報D 2 1に基づいてALU 1 2
と入力用のセクタ5 9とを接続する。信号線L 2 1には演算結果値Zが伝送さ
れる。信号線L 2 2は配線情報D 2 2に基づいてALU 1 2とラッチ回路5 1 0
とを接続する。信号線L 2 2には一致検出信号S 2 2が伝送される。信号線L 2
20 3は配線情報D 2 3に基づいて実行ステートマシーン5 1とラッチ回路5 1 0と
を接続する。信号線L 2 3にはフラグ状態信号S 2 3が伝送される。信号線L 2
4は配線情報D 2 4に基づいて入力用のセクタ5 9と実行ステートマシーン5
1とを接続する。信号線L 2 4には選択制御信号S 2 4が伝送される。

信号線L 2 5は配線情報D 2 5に基づいて入力用のセクタ5 9とレジスタア
25 レイ1 1とを接続する。信号線L 2 5にはデータ、演算結果値Z、被数X値等が
伝送される。信号線L 2 6は配線情報D 2 6に基づいて実行ステートマシーン5
1と命令読出しステートマシーン5 2とを接続する。信号線L 2 6には実行終了
信号S 2 6が伝送される。信号線L 2 7は配線情報D 2 7に基づいて実行ステ
ートマシーン5 1とセクタ5 3とを接続する。信号線L 2 7には分岐制御信号S

27が伝送される。信号線L28は配線情報D28に基づいて実行ステートマシーン51とセクタ53とを接続する。信号線L28には選択制御信号S28が伝送される。

- 信号線L29は配線情報D29に基づいて命令読出しステートマシーン52と
- 5 実行ステートマシーン51とを接続する。信号線L29には命令実行開始信号S29が伝送される。信号線L30は配線情報D30に基づいて命令読出しステートマシーン52とプログラムカウンタ54とを接続する。信号線L30にはカウンタ制御信号S30が伝送される。信号線L31は配線情報D31に基づいてセクタ57と実行ステートマシーン51とを接続する。信号線L31には選択制御信号S31が伝送される。信号線L32は配線情報D32に基づいてセクタ56と実行ステートマシーン51とを接続する。信号線L32には選択制御信号S32が伝送される。アドレス信号線L33は配線情報D33に基づいてセクタ56と実行ステートマシーン51とを接続する。アドレス信号線L33には読出しアドレスA_rが伝送される。信号線L34は配線情報D34に基づいてラッチ回路58と実行ステートマシーン51とを接続する。信号線L34にはラッチ制御信号S58が伝送される。
- 10
- 15

- 信号線L35は配線情報D35に基づいてALU12と実行ステートマシーン51とを接続する。信号線L35にはALU制御信号S35が伝送される。信号線L36は配線情報D36に基づいてラッチ回路58とALU12とを接続する。
- 20 信号線L36にはラッチ出力としてX値「0」等が伝送される。アドレス信号線L37は配線情報D37に基づいてセクタ57と実行ステートマシーン51とを接続する。アドレス信号線L37には書込みアドレスA_wが伝送される。信号線L38は配線情報D38に基づいて実行ステートマシーン51とラッチ回路510とを接続する。信号線L38にはラッチ制御信号S38が伝送される。

- 25 このように、本発明に係る第2の実施例としての演算処理装置100の構築方法によれば、電源オンと共にフラッシュメモリ70から読み出された配線情報D1～D38・・・等に基づいてレジスタアレイ11、ALU12、デコーダ13、ROMセル14、実行ステートマシーン51、命令読出しステートマシーン52、セクタ53、プログラムカウンタ54、インクリメンタ55、セクタ56、

セクタ 57、入力用のセクタ 59 及びラッチ回路 58、510、511 を結線するようになされる。従って、実行ステートマシーン 51 で演算命令 #I1 ~ #I4 に基づいて一のレジスタ r10 を選択し、このレジスタ r10 が保持する値「0」によって他のレジスタ r0 を選択するレジスタ相対レジスタアドレッシング処理を実行するようなプログラム可能な演算処理装置 100 を電源オンと共に構築することができる。1 チップでメモリセルを含むプロセッサを効率良く構築することが可能となる。

この装置構築後のデコーダ 13 では ALU12 を動作させるための演算プログラム AP から演算命令 #I1 ~ #I4 が解読され、命令実行制御部 50 ではデコーダ 13 によって解読された演算命令 #I1 ~ #I4 を実行するためにレジスタアレイ 11 及び ALU12 が制御される。ALU12 ではレジスタアレイ 11 から読み出された X 値、Y 値が演算される。そのレジスタ r10 が示す値「0」のレジスタ r0 にその演算結果値 Z を格納することができる。

また、命令実行制御部 50 では PLD 内部のメモリ機能を持つ部分の全てをレジスタとして扱うことができるので、メモリセルとレジスタを区別してアクセスする種類のプロセッサと比べて、高速な動作が可能になる。しかも、CPU や、RAM、ROM 等を個々に基板に配置する場合に比べて基板占有面積を低減することができる。当該演算処理装置 100 が適用される携帯端末装置等の電子機器のコンパクト化を図ることができる。

20 (3) 第 3 の実施例

図 13 は本発明に係る第 3 の実施例としての演算処理システム 100' の構成例を示すブロック図である。

この実施例では複数のレジスタを指定してレジスタ相対レジスタアドレッシング処理を実行する演算プログラムに基づいて演算処理を実行する演算処理装置 (第 2 の演算処理装置) 100' を備え、プログラム作成系で作成された圧縮プログラムを取得してレジスタ種類を解読し、このレジスタ種類に基づいて当該レジスタを指定する命令ビット数を復元し、所定の命令長の命令構造を有するプログラムを復元するようにして、プログラムデータを格納する ROM 等のメモリ容量を低減できるようにすると共に、メモリセルや論理演算素子から成る PLD に

- よりプロセッサを構築する場合に、ROMとして機能させるメモリセルの占有率を低減できるようにしたものである（第2の演算処理方法）。また、プログラム作成系ではレジスタの使用頻度に応じて命令の長さを可変できるようにすると共に、頻繁にアクセスするレジスタには短い長さの命令をセットできるようにする（第3の演算処理方法）。

- 図13に示す演算処理システム10は、一方で、所定のプログラム言語に基づいてレジスタ相対レジスタアドレッシング処理を実行する演算プログラムに基づいて演算処理を実行するための命令を編集してプログラムを作成し、他方で、当該プログラムと複数のレジスタとを使用してデータを処理するシステムである。
- 10 演算処理システム10ではプログラム作成系Iを成すプログラム作成装置200が準備される。新規に設計製造される演算処理装置100'を動作させるためのプログラムを作成するためである。演算処理装置100'はプログラム実行系IIを構成し、当該装置100'内には命令実行演算部や、記憶部、複数のレジスタ等が実装される。プログラム作成装置200では、演算処理装置100'でレジスタを使用する頻度に基づいて当該レジスタを指定するための命令ビット数を圧縮すると共に、当該プログラムの命令構造の中にレジスタ種類を記述して命令長の異なる圧縮プログラムAPを作成するようになされる。演算処理装置100'に実装されるプログラム格納用の記憶部のメモリ容量を削減するためである。

- プログラム作成装置200は例えば、データベース21、キーボード22、マウス23、表示装置24及び制御装置25を有している。データベース21には演算処理装置100'のプログラム作成に必要なデータが格納される。例えば、C言語によるプログラムの記述に必要な「Global変数宣言」、「関数宣言」、「Local変数宣言」、「代入」、「加算」、「比較」及び「分岐」が格納される。データベース21には制御装置25が接続されており、この制御装置25にはキーボード22、マウス23及び表示装置24が接続されている。プログラム作成装置200では表示装置24にC言語によるプログラム記述画面を表示して、キーボード22及びマウス23を使用してプログラムが作成される。

例えば、新規な設計製造に係る演算処理装置100'がN個のレジスタを使用する場合であって、N個のレジスタに第1番から第N番のシリアル番号を付与し

たとき、第1番から第K番のグループのレジスタを使用頻度が高い部類として「Local変数宣言」がなされ、第K+1番から第N番のグループのレジスタを使用頻度が低い部類として「Global変数宣言」がなされる。これらの宣言はキーボード22や、マウス23を使用して指定され、これはレジスタ種類を2つに分類して使用頻度が高いレジスタは短い命令ビット数で命令セットし、使用頻度が低いレジスタは長い命令ビット数で命令セットするためである。演算処理装置100'においてレジスタの数が例えば、4千~8千個程度になると、これを指定する命令ビット数が12乃至13ビット必要になる。

制御装置25ではレジスタの使用頻度に応じて命令の長さを可変するようになされる。頻繁にアクセスするレジスタは短い長さの命令をセットするためである。使用頻度が高いレジスタは短い命令ビット数が割り当てられ、使用頻度が低いレジスタは長い命令ビット数が割り当てられる。演算処理装置100'ではこのプログラム作成装置200で作成された圧縮プログラムAPを取得してレジスタ種類を解読し、このレジスタ種類に基づいて当該レジスタを指定する命令ビット数を復元し、所定長さの命令に基づいて複数のレジスタを指定して任意の演算を実行するようになされる。

演算処理装置100'は例えば、命令解読復元部3、記憶部4、レジスタアレイ11'及び命令実行演算部50'を有している。レジスタアレイ11'は複数のレジスタを集合したものである。記憶部4にはレジスタアレイ11'の中から該当レジスタを指定するための圧縮プログラムAPが格納される。圧縮プログラムAPはプログラム作成装置200で作成されたものが使用される。例えば、圧縮プログラムAPは演算処理装置100'で構築されたプログラム格納用の記憶部4に、ROMライタ等を使用して書き込まれる。これは演算処理装置100'に関して、複数のメモリセルや算術論理素子により構成されるプログラマブル・ロジック・デバイス(Programmable Logic Device; PLD)からプロセッサを構築する場合があるからである。ROMとして機能させるメモリセルの占有率を低減することができる。もちろん、プログラム格納用の記憶部4を演算処理装置100'とは別個に製造し、個々の記憶部4に圧縮プログラムAPを格納してから同一基板上に実装する方法であってもよい。記憶部4として読み出し専用の

メモリ（ROM）や、EEPROM（フラッシュメモリ）が使用されるからである。

記憶部4には命令解読復元部3が接続されており、この記憶部4から圧縮プログラムAPを読み出してレジスタ種類を解読し、このレジスタ種類に基づいて当該レジスタを指定するための命令ビット数を復元するようになされる。命令長を揃え、この命令に基づいて複数のレジスタを指定するためである。命令解読復元部3には命令実行演算部50'が接続されており、この命令実行演算部50'にはレジスタアレイ11'が接続されている。命令実行演算部50'では命令解読復元部3によって復元された所定の命令長のプログラムに基づいてレジスタアレイ11'内で該当レジスタを指定して任意の演算を実行するようになされる。

続いて、本発明に係る演算処理方法について、当該演算処理システム10における処理例について説明をする。図14は演算処理システム10における処理例を示すフローチャートである。このシステム10ではプログラム作成系Iで所定のプログラム言語に基づいてレジスタ相対レジスタアドレッシング処理を実行する演算プログラムに基づいて演算処理を実行するための命令を編集してプログラムを作成し、プログラム実行系IIで当該プログラムと複数のレジスタとを使用してデータを処理する場合を前提とする。この演算処理装置100'がN個のレジスタを使用する場合であって、N個のレジスタに第1番から第N番のシリアル番号が付与される場合を例にとる（第2の演算処理方法）。

これを処理条件にして、プログラム作成系では図14AにフローチャートのステップA01で所定のプログラム言語に基づいてレジスタ相対レジスタアドレッシング処理を実行する演算プログラムに基づいて演算処理を実行するための命令を編集する。そして、ステップA02でレジスタを使用する頻度に基づいて当該レジスタを指定する命令ビット数を圧縮し、命令長を短くする。例えば、第K+1番から第N番のグループのレジスタを指定する命令ビット数をnビットとし、第1番から第K番のグループのレジスタを指定する命令ビット数をmビットとしたとき、例えば、 $n - m = 8$ ビットとなるように、第1番から第K番のグループのレジスタを指定する命令ビット数が圧縮される。

その後、ステップA03で当該プログラムの命令構造の中にレジスタ種類を記

- 述する。例えば、被数及び加数を保持する1組のレジスタ、これを「レジスタ番号1」のレジスタの種類を「レジスタ種類1」、及び「レジスタ番号2」のレジスタの種類を「レジスタ種類2」としたとき、第1番から第K番のグループのレジスタに関して使用頻度が高い場合は「レジスタ種類1」及び「レジスタ種類
- 5 2」にコード「0」が記述される。また、第K+1番から第N番のグループのレジスタに関して使用頻度が低い場合は「レジスタ種類1」及び「レジスタ種類2」にコード「1」が記述される。

- そして、ステップA04で命令長の異なる圧縮プログラムAPを作成する。この圧縮プログラムAPにおいて、第1番から第K番のグループのレジスタを指定
- 10 する命令ビット数に関してはmビットであり、第K+1番から第N番のグループのレジスタを指定する命令ビット数に関してはnビットである。上述の例で第1番から第K番のグループのレジスタ指定を含む命令形態では、第K+1番から第N番のグループのレジスタ指定を含む命令形態に比べて命令長が16ビット短くなる。

- 15 一方、プログラム実行系ではプログラム作成系で作成された圧縮プログラムAPを図14Bに示すフローチャートのステップB1で取得する。例えば、圧縮プログラムAPは演算処理装置100'で構築されたプログラム格納用の記憶部4に、ROMライタ等を使用して書き込まれる。この圧縮プログラムAPでは使用頻度が高いレジスタは短い命令ビット数=mビットで命令セットされ、使用頻度
- 20 が低いレジスタは長い命令ビット数=nビットで命令セットされている。そして、ステップB2で命令を実行するかを判断する。この際の判断は周知技術によりなされる。命令を実行する場合はステップB3でレジスタ種類を解読する。例えば、「レジスタ種類1」及び「レジスタ種類2」に関してコード「0」から使用頻度が高いレジスタとして第1番から第K番のグループのレジスタ番号が解読され、
- 25 「レジスタ種類1」及び「レジスタ種類2」に関してコード「1」から使用頻度が低いレジスタとして第K+1番から第N番のグループのレジスタ番号が解読される。

そして、解読されたレジスタ種類に基づいてステップB4で当該レジスタを指定する命令ビット数を復元する。例えば、第1番から第K番のグループのレジス

タの命令ビット数=mビットの上位、この例で上位8ビットに「0」が付加される。第1番から第K番のグループのレジスタの命令ビット数が、第K+1番から第N番のグループのレジスタの命令ビット数と同様にしてnビットに揃えられる。ここで復元された所定長さの命令に基づいてステップB5で複数のレジスタを指定して任意の演算を実行する。その後、ステップB6で演算処理を終了するかを判断する。演算処理を終了しない場合はステップB2に戻って命令を実行するかを判断して演算処理を継続する。演算処理を終了する場合は電源オフ情報等を検出して当該演算処理を終了する。

このように、本発明に係る第3の実施例としての演算処理システム10によれば、一方で、所定のプログラム言語に基づいてレジスタ相対レジスタアドレッシング処理を実行する演算プログラムに基づいて演算処理を実行するための命令を編集してプログラムを作成し、他方で、当該プログラムと複数のレジスタとを使用してデータを処理する場合に、プログラム作成装置200ではレジスタを使用する頻度に基づいて当該レジスタを指定する命令ビット数を圧縮すると共に、当該プログラムの命令構造の中にレジスタ種類を記述して命令長の異なる圧縮プログラムAPが作成される。演算処理装置100'では、プログラム作成装置200で作成された圧縮プログラムAPを取得してレジスタ種類を解読し、このレジスタ種類に基づいて当該レジスタを指定する命令ビット数を復元し、所定長さの命令に基づいて複数のレジスタを指定して任意の演算が実行される。従って、プログラム作成系Iではレジスタの使用頻度に応じて命令の長さを可変できるので、頻繁にアクセスするレジスタに短い長さの命令をセットすることができる。これにより、プログラム実行系IIではROM等の記憶部4に圧縮した命令をセットすることができ、プログラムデータを格納する記憶部4のメモリ容量を低減することができる。また、メモリセルや論理演算素子から成るPLDによりプロセッサを構築する場合に、ROMとして機能させるメモリセルの占有率を低減することができ、その分のメモリセルをレジスタに多く割り当てることができるようになる。

(4) 第4実施例

図15は本発明に係る第4の実施例としてのマイクロプロセッサ101の構成

例を示すブロック図である。

この実施例では演算処理装置 100' に外部メモリ 2 を接続してマイクロプロセッサ 101 を構成し、複数のレジスタを指定してレジスタ相対メモリアドレス
5 ング処理を実行する演算プログラムに基づいて演算処理を実行する。もちろん、
第 1 の実施例と同様にして、レジスタ相対レジスタアドレスング処理を実行す
るための演算命令を圧縮プログラム AP に記述してもよい。この場合には、図 1
5 に示すレジスタアレイ 11' の入力部分に図 1 に示したようなセクタ 56,
57 を接続し、このセクタ 56, 57 を実行ステートマシン 51 で制御すれば
よい。第 1 の実施例の効果と第 2 の実施例の効果とが重複して得られる。

10 この例では、プログラム作成系 I で作成された機械語の命令の圧縮プログラム
AP を取得してレジスタ種類を解読し、このレジスタ種類に基づいて当該レジス
タを指定する命令ビット数を復元し、所定の命令長の命令構造を有するプログラ
ムを復元するようにした。そうすることでプログラムデータを格納する ROM 等
のメモリ容量を低減できるようにしたものである。

15 図 15 に示すマイクロプロセッサ 101 は、レジスタを使用する頻度に基づい
て当該レジスタを指定する命令ビット数が予め圧縮されると共に、当該プログラ
ムの命令構造の中にレジスタ種類が記述された命令長の異なる圧縮プログラム A
P に基づいてデータを処理する装置である。このプロセッサ 101 ではプログラ
ム作成装置 200 で作成された圧縮プログラム AP を取得してレジスタ種類を解
20 読し、このレジスタ種類に基づいて当該レジスタを指定する命令ビット数を復元
し、所定長さの命令に基づいて複数のレジスタを指定して任意の演算を実行する
ようになされる。

マイクロプロセッサ 101 は例えば、レジスタアレイ 11'、命令ビット復元
デコーダ（第 1 の実施例のデコーダに相当）13、ROM 14、及び命令実行演
25 算部 50' を有している。レジスタアレイ 11' は複数のレジスタを集合したも
のである。レジスタアレイ 11' には例えば、8192 個×32 bit のレジス
タ r_i ($i=0\sim 8191$) が設けられる。各々のレジスタ r_i は書込みアドレ
ス A_w 及び書込み制御信号 S_w に基づいて任意の値を保持し、及び、読出しアド
レス A_r に基づいて被数 X や加数 Y 等の値を出力するようになされる。

このマイクロプロセッサ 101 は記憶部の一例となるプログラム格納用の ROM 14 が実装されており、レジスタアレイ 11' の中から該当レジスタ r_i を指定するための圧縮プログラム AP が格納される。圧縮プログラム AP は機械語の命令 (Instruction) 構造を有しており、プログラム作成装置 200 で作成されたものが使用される。例えば、圧縮プログラム AP は ROM ライタ等を使用して ROM 14 に書き込まれる。命令実行時、ROM 14 は例えば、プログラムカウンタ 54 からのカウント出力信号 S5 に基づいて圧縮プログラム AP を出力するようになされる。

ROM 14 には命令ビット復元デコーダ 13 が接続されており、この ROM 14 から機械語の命令の圧縮プログラム AP を読み出して命令制御信号 S4、命令信号 S9 及び各引数信号 S10 を発生するようになされる。命令信号 S9 には load 命令、add 命令、cmp 命令、jump 命令が含まれる。各引数信号 S10 にはアクセス方法 #1、アクセス方法 #2、「レジスタ種類 1」、「レジスタ種類 2」、レジスタ番号 $r_0, r_1 \dots$ 等、フラグ状態 (flag condition) 及びジャンプアドレス等が含まれる。

この圧縮プログラム AP にはレジスタ相対メモリアドレッシング処理を実行するための演算命令を含んでいる。この処理では演算命令に基づいて一のレジスタを選択し、ここで選択されたレジスタが保持する値によって外部メモリ 2 を選択するようになされる。この処理は例えば、アクセス方法 #1 によって実行される。

この例で命令ビット復元デコーダ 13 は、「レジスタ種類 1」及び「レジスタ種類 2」を解読し、この「レジスタ種類 1」及び「レジスタ種類 2」に基づいて当該レジスタ r_i を指定するための命令ビット数 = n ビットを復元するようになされる。命令長を圧縮前のビット数に揃え、この命令に基づいて複数のレジスタ r_i 等を指定するためである。上述の命令制御信号 S4 は命令読出しステートマシン 52 に出力される。上述のレジスタアレイ 11' 及び命令ビット復元デコーダ 13 には命令実行演算部 50' が接続されている。命令実行演算部 50' では命令ビット復元デコーダ 13 によって復元された所定の命令長のプログラムに基づいてレジスタアレイ 11' 内で該当レジスタ r_i を指定して任意の演算を実行するようになされる。

命令実行演算部 50' は、算術論理演算ユニット (Arithmetic and Logic Unit: 以下でALUという) 12、実行ステートマシーン 51、命令読出しステートマシーン 52、セクタ 53、プログラムカウンタ (PC) 54、+1インクリメンタ 55、入力用のセクタ 59、ラッチ回路 58、510、511' を有しており、レジスタ相対メモリアドレッシング処理を実行するようになされる。レジスタアレイ 11' にはデータ信号線 L20 が接続されており、この信号線 L20 を通じて ALU 12 が接続されている。ALU 12 ではレジスタアレイ 11' の中で指定されたレジスタから読み出された X や Y 等の値を演算するようになされる。演算結果の値は Z である。演算種目は足し算、かけ算、引き算、わり算等である。演算種目は実行ステートマシーン 51 から出力される ALU 制御信号 S35 に基づいて設定される。データ信号線 L20 には、ALU 12 の他にラッチ回路 58、511、59 等が接続されている。データ信号線 L20 には DATA、被数 X 値、加数 Y 値等が伝送される。

命令ビット復元デコーダ 13 には実行ステートマシーン 51 及び命令読出しステートマシーン 52 が接続されており、命令ビット復元デコーダ 13 によって解読された演算命令を実行するためにレジスタアレイ 11' 及び ALU 12 を制御するようになされる。命令読出しステートマシーン 52 では命令ビット復元デコーダ 13 から出力される命令制御信号 S4 に基づいてプログラムカウンタ 54 及び実行ステートマシーン 51 を制御する。例えば、当該マシーン 52 は命令ビット復元デコーダ 13 から命令信号 S9 及び各引数信号 S10 が実行ステートマシーン 51 へ出力されると共に命令実行開始信号 S29 を出力する。

実行ステートマシーン 51 にはレジスタアレイ 11'、ALU 12、ラッチ回路 58、510、511 及びセクタ 59 が接続されている。当該マシーン 51 では命令実行開始信号 S29 に基づいて命令の実行を開始する。例えば、データの書込み時には、書込み制御信号 S_w がレジスタアレイ 11' に出力され、セクタ 59 には選択制御信号 S24 が出力される。データの読出し時には、読出しアドレス A_r がレジスタアレイ 11' に出力される。演算時には、ラッチ制御信号 S34 がラッチ回路 58 に出力され、ラッチ回路 510 にはラッチ制御信号 S38 が出力される。当該プロセッサ外部には外部制御信号 S16 が出力される。

命令の実行が終了すると、実行ステートマシーン 5 1 は命令読み出しステートマシーン 5 2 へ実行終了信号 S 2 6 を出力し、プログラムカウンタ 5 4 の値を進めるようになされる。

- この実行ステートマシーン 5 1 及び命令読み出しステートマシーン 5 2 にはセ
- 5 レクタ 5 3 が接続されており、選択制御信号 S 2 8 に基づいてインクリメント出力信号 S 7 又は分岐制御信号 S 2 7 のいずれか一方を選択し、これをセクタ出力としてプログラムカウンタ 5 4 に出力するようになされる。選択制御信号 S 2 8 は実行ステートマシーン 5 1 から供給される。インクリメント出力信号 S 7 はインクリメンタ 5 5 からセクタ 5 3 へ出力される。プログラムカウンタ 5 4 で
- 10 はカウント制御信号 S 3 0 に基づいて ROM 1 4 から圧縮プログラム AP を読み出す場所が指定される。+1 インクリメンタ 5 5 はプログラムカウンタ 5 4 のカウント出力信号 S 5 を「+1」してインクリメントするようになされる。カウント制御信号 S 3 0 は命令解読ステートマシーン 5 2 から供給される。このカウント出力信号 S 5 は+1 インクリメンタ 5 5 の他に ROM 1 4 に出力される。
- 15 セクタ 5 9 はデータバス 1 9 A、レジスタアレイ 1 1' 及び ALU 1 2 に接続されており、データバス 1 9 A から取り込んだデータ (DATA)、レジスタアレイ 1 1' から出力される被数 X 値 (加数 Y 値) 又は ALU 1 2 から出力される演算結果値 Z のいずれかを選択制御信号 S 2 4 に基づいて入力制御するようになされる。ラッチ回路 5 8 はレジスタアレイ 1 1' の読出しポートと ALU 1 2 の間に
- 20 に接続されており、ラッチ制御信号 S 3 4 に基づいてレジスタ r i の出力値 X をラッチするようになされる。ラッチ回路 5 1 0 は ALU 1 2 の比較出力部等に接続されており、ラッチ制御信号 S 3 8 に基づいて一致検出信号 S 2 2 をラッチして、フラグ状態 (flag condition) 信号 S 2 3 を出力するようになされる。ラッチ回路 5 1 1 はレジスタアレイ 1 1' の読出しポートとアドレスバス 1 9 B との
- 25 間に接続されており、ラッチ制御信号 S 1 7 に基づいて外部アドレス (address) をラッチするようになされる。

なお、実行された命令によってジャンプ (命令分岐) が発生した場合は、ジャンプ先のアドレスを示す分岐制御信号 S 2 7 を実行ステートマシン 5 1 からセクタ 5 3 へ出力される。セクタ 5 3 では選択制御信号 S 2 8 に基づいてその分

岐制御信号S 2 7を選択し、この分岐制御信号S 2 7をプログラムカウンタ5 4へ書き込むようになされる。また、実行ステートマシーン5 1、セクタ5 9、データ信号線L 2 0及びラッチ回路5 1 1にはI/Oインタフェース6 0を通じて外部メモリ2が接続されている。レジスタ相対メモリアドレッシング処理に基づいてALU1 2を動作させるためである。I/Oインタフェース6 0と外部メモリ2との間はデータバス1 9 A、アドレスバス1 9 B及びコントロールバス1 9 Cによって接続され、データバス1 9 Aによってデータが転送され、アドレスバス1 9 Bによってアドレスが転送され、コントロールバス1 9 Cによって外部制御信号S 1 6が外部メモリ2へ転送される。外部メモリ2を制御するためである。外部メモリ2には例えば、5 1 2 Mバイト×3 2 ビットのRAM（随時書き込み読み出し可能なメモリ）が使用される。

図1 6はレジスタアレイ1 1'の内部構成例を示すブロック図である。図1 6に示すレジスタアレイ1 1'によれば、例えば、8 1 9 2個の3 2 b i tのレジスタ r_i ($i=0\sim 8 1 9 1$)が備えられ、各々のレジスタ r_i の入力には書き込みポート1 5が接続されている。1 b i tのレジスタはD型のフリップ・フロップ回路等から構成される。書き込みポート1 5は図1 5に示したセクタ5 9に接続されており、書き込み制御信号 S_w 及び書き込みアドレス A_w に基づいて、データバス1 9 Aから取り込んだデータ（DATA）、レジスタアレイ1 1'から出力される被数X値（加数Y値）又はALU1 2から出力される演算結果値Zのいずれかをレジスタ $r_0\sim r_i$ に書き込むようになされる。書き込みポート1 5は実行ステートマシーン5 1に接続され、書き込みアドレス A_w を供給するようになされる。各々のレジスタ r_i の出力には読出しポート1 6が接続されている。読出しポート1 6はデータ信号線L 2 0を通じて図1 5に示したALU1 2、ラッチ回路5 8、5 1 1、セクタ5 9等に接続されており、読出しアドレス A_r に基づいて指定されたレジスタ r_i からデータ（DATA）を読み出すようになされる。読出しポート1 6は実行ステートマシーン5 1に接続され、読出しアドレス A_r を供給するようになされる。

次に、図1 7 A～E及び図1 8 A～図1 8 Dを参照しながら、ROM1 4にセットされる命令の構造例について説明する。先に説明したROM1 4にセットさ

れる命令形態は、図17A～図17Eに示すように、#F1～#F5の5種類である。マイクロプロセッサ101では命令形態#F2～#F4の命令を命令形態#F1の命令の形態に復元して取り扱われる。これらの命令形態#F1～#F5で各々の命令はload、add及びcmp命令と、jump命令とに大きく2つに分かれる。命令形態#F1～#F4において、load命令の場合は図18Aに示すように命令にコード「0」が記述され、add命令の場合は命令にコード「1」が記述され、cmp命令の場合は命令にコード「2」が各々記述される。命令形態#F5はjump命令であって、この場合は命令にコード「3」が記述される。cmp命令に関しては比較結果が同じであった場合は、図15に示したラッチ回路510のフラグ状態信号S23に基づいてzero flagが「1」にセットされ、同じでなかった場合は「0」がセットされる。

この例では、load、add及びcmp命令はアクセスするレジスタriの番号が例えば、5ビットで表現できるときは5ビット、それ以外は13ビットで表現される。つまり、使用頻度が高い第0番のレジスタr0から第31番のレジスタr31をm=5ビットで表現する。使用頻度が低い第32番のレジスタr32から第8191番のレジスタr8191をm=13ビットで表現するようになされる。命令形態#F1では命令長が32ビットであり、「レジスタ番号1」で示されるレジスタriの命令ビット数はn=13ビットであり、「レジスタ番号2」で示されるレジスタriの命令ビット数もn=13ビットである。命令形態#F2では命令長が24ビットであり、「レジスタ番号1」で示されるレジスタriの命令ビット数はm=5ビットであり、「レジスタ番号2」で示されるレジスタriの命令ビット数はn=13ビットである。

また、命令形態#F3でも命令長が24ビットであり、「レジスタ番号1」で示されるレジスタriの命令ビット数はn=13ビットであり、「レジスタ番号2」で示されるレジスタriの命令ビット数はm=5ビットである。命令形態#F4は命令長が16ビットであり、「レジスタ番号1」で示されるレジスタriの命令ビット数はm=5ビットであり、「レジスタ番号2」で示されるレジスタriの命令ビット数もm=5ビットである。いずれの命令形態#F1～#F4においても、最初の2ビットは命令の種類を示している。命令種類に関して、lo

a d は転送、a d d は加算、c m p は比較、j u m p は制御移行（分岐）を各々示している。l o a d、a d d、c m p 命令の場合は、命令の後に2ビットのアクセス方法#1、#2が続く。Operandは左がアクセス方法#1、「レジスタ番号1」で表され、右がアクセス方法#2、「レジスタ番号2」で表される。

- 5 つまり、アクセス方法#1は「レジスタ番号1」で示されるレジスタ r_i のアクセス方法を示しており、アクセス方法#2は「レジスタ番号2」で示されるレジスタ r_i のアクセス方法を示している。アクセス方法#1とアクセス方法#2はそれぞれレジスタ番号No. 1、レジスタ番号No. 2に対応し、これらの間で処理が行われる。いずれも、図18Bに示すようにアクセス方法#1、#2には2種類が準備されている。アクセス方法#1、#2に関してコード「0」が記述された場合は、レジスタ番号2で示されるレジスタ r_i に対し直接アクセスする方法である。当該レジスタ番号で示されるレジスタ r_i の値を直接用いることを示している。アクセス方法#1、#2に関してコード「1」が記述された場合は、「レジスタ番号1」で示されるレジスタ r_i の値をアドレスとし、当該マイクロプロセッサ101で外部メモリ2に対してアクセスする方法である（図18B参照）。
- 10
- 15

- また、図17A～図17Dにおいて、アクセス方法#1、#2の後には2ビットの「レジスタ種類1」、「レジスタ種類2」が続けて記述されている。「レジスタ種類1」は「レジスタ番号1」で示されるレジスタ r_i の種類を示し、「レジスタ種類2」は「レジスタ番号2」で示されるレジスタ r_i の種類を示している。レジスタ種類は図18Cに示すように2種類が準備されている。「レジスタ種類1」及び「レジスタ種類2」に関して、コード「0」が記述される場合は、レジスタ番号が「31」以下で使用頻度が高いレジスタ r_i を示している。このレジスタ r_i ($i=0\sim31$) はレジスタ番号を $m=5$ ビットで表現することができる。この「レジスタ種類1」及び「レジスタ種類2」に関して、「1」が記述される場合は、レジスタ番号が「32」以上で使用頻度が低いレジスタ r_i を示している。このレジスタ r_i ($i=32\sim8191$) はレジスタ番号を $n=13$ ビットで表現するようになされる。このように、レジスタ番号の大きさを区別することでプログラムを圧縮することができる。この「レジスタ種類1」、「レ
- 20
- 25

レジスタ種類 2」の後には「レジスタ番号 1」、「レジスタ番号 2」が続けて記述されている。「レジスタ番号 1」は例えば、被数を保持するレジスタ r_i を示し、「レジスタ番号 2」は加数を保持するレジスタ r_i を示す。

また、図 17 E に示す `jump` 命令のフォーマットによれば、最初の 2 bit
5 に命令が記述され、続く 2 bit にはフラグ状態 (flag condition) が記述される。続く 20 bit にはジャンプアドレスが記述される。フラグ状態は図 18 D に示すように、命令実行制御を移すかどうかの判断をするための条件である。コード「0」は「無条件」で常に制御を移す。コード「1」は「zero flag」で zero flag が「1」である場合に、制御を移す。コード「2」は「non-zero
10 flag」で zero flag が「0」である場合に制御を移すようになされる。コード「3」は未使用である。

続いて、プログラム作成系 I における処理例について説明をする。図 19 はプログラム作成系 I におけるプログラム作成例を示す表図である。図 19 において、P 1 はプログラム記述画面をイメージし、P 2 にはその記述内容を示し、P 3 には
15 は実施例で該当する条件を記述したものである。所定のプログラム言語に基づいてレジスタ相対メモリアドレッシング処理を実行する演算プログラムに基づいて演算処理を実行するための命令を、図 1 に示したプログラム作成装置 200 で編集して圧縮プログラムを作成するためである。

プログラム作成装置 200 では図 1 に示した表示装置 24 に図 19 に示す C 言語
20 によるプログラム記述画面 P 1 を表示して、キーボード 22 及びマウス 23 を使用して圧縮プログラムが作成される。このとき、データベース 21 からマイクロプロセッサ 101 のプログラム作成に必要なデータが読み出される。例えば、C 言語によるプログラムの記述に必要な「Global 変数宣言」、「関数宣言」、「Local 変数宣言」、「代入」、「加算」、「比較」及び「分岐」が読み出される。
25 この例で新規な設計製造に係るマイクロプロセッサ 101 で、 $N=8192$ 個の 32 ビットのレジスタ r_i を使用する場合であって、8192 個のレジスタ r_i に第 0 番から第 8191 番のシリアル番号を付与したとき、第 0 番から第 31 番のグループのレジスタ $r_0 \sim r_{31}$ を使用頻度が高い部類として「Local 変数宣言」がなされる。

つまり、この例では第31番目以下のレジスタ r_i に「Local変数宣言」が割り当てられる。また、第32番から第8191番のグループのレジスタ $r_{32} \sim r_{8191}$ を使用頻度が低い部類として「Global変数宣言」がなされる。つまり、第32番目以上のレジスタ r_i には「Global変数宣言」が割り当てられる。プログラム作成装置200では「Global変数宣言」がなされた第32番から第8191番のグループのレジスタ $r_{32} \sim r_{8191}$ の命令ビット数を $n=13$ ビットとしたとき、Local変数宣言がなされたレジスタ $r_0 \sim r_{31}$ の命令ビット数は、これよりも8ビット少ない $m=5$ ビットに圧縮される。これと共に、当該プログラムの命令構造の中に「レジスタ種類1」、「レジスタ種類2」を記述して命令長の異なる圧縮プログラムAPを作成するようになされる。

図13に示した制御装置25ではレジスタ r_i の使用頻度に応じて命令の長さを可変するようになされる。この例では、使用頻度が高いレジスタ $r_0 \sim r_{31}$ を短い命令ビット数 $m=5$ ビットで命令セットし、使用頻度が低いレジスタ $r_{32} \sim r_{8191}$ は長い命令ビット数 $n=13$ ビットで命令セットするようになされる。このようにすると、頻繁にアクセスするレジスタ $r_0 \sim r_{31}$ は短い長さの命令をセットすることができ、マイクロプロセッサ101に実装されるプログラム格納用のROM14のメモリ容量を削減することができる。

続いて、プログラム作成装置200におけるコンパイル例について説明をする。図20はプログラム作成装置200におけるコンパイル例を示すフローチャート（メインルーチン）である。図21はコンパイラにおける代入及び演算処理例を示すフローチャート（サブルーチン）である。

この実施例ではプログラム作成系IでC言語によるプログラムに基づいてレジスタ相対メモリアドレス処理を実行する演算プログラムに基づいて演算処理を実行するための命令を編集して圧縮プログラムを作成する場合を前提とする。また、マイクロプロセッサ101が8192個のレジスタ $r_0 \sim r_{8191}$ を使用する場合であって、8192個のレジスタ r_i に第0番から第8191番のシリアル番号が付与される場合を例にとる。これを処理条件にして、プログラム作成系IではC言語によるプログラムに基づいて命令を編集するために、図20にフローチャートのステップC1でプログラムアドレスを「0」にする。その後、

ステップC 2に移行してC言語によるプログラムを一行読み込む。このとき、表示装置24のプログラム記述画面P1には、例えば、「global変数宣言」を示す

```
int *read__add *write__add, counter, end__val ;
```

が表示され、また、関数宣言を示す

```
5 void main ()
```

```
{
```

が表示される。

そして、ステップC 3で当該プログラムの記述が「global変数宣言」であるかをチェックする。当該記述が「global変数宣言」である場合は、ステップC 4に移行して第32番目以上のレジスタr 32～r 8191を割り当てる。このグループのレジスタr 32～r 8191を指定する命令ビット数をmビットとしたとき、m=13ビットである。命令は命令形態#F1で作成される。その後、ステップC 14に移行する。また、ステップC 3で当該記述が「global変数宣言」ではない場合はステップC 5に移行して「local変数宣言」かをチェックする。この

15 とき、表示装置24のプログラム記述画面P1には、例えば、「local変数宣言」を示す

```
int temp, added__val ;
```

が表示される。当該記述が「local変数宣言」である場合は、ステップC 6に移行して第31番目以下のレジスタr 0～r 31を割り当てる。レジスタr iを使用

20 する頻度が高いことから、当該レジスタr iを指定する命令ビット数nを、「global変数宣言」されたレジスタr 32～r 8191よりも8ビット少ない5ビットに圧縮するためである。命令は命令形態#F2乃至#F4で作成される。その後、ステップC 14に移行する。

上述のステップC 5で当該記述が「local変数宣言」ではない場合はステップC

25 7に移行して、C言語のプログラムで代入・加算処理等の実行を示す「do」が記述されているかをチェックする。このとき、表示装置24のプログラム記述画面P1には、例えば、「do」を示す

```

do {
    temp = *read__add;
    temp = temp + added__val;
    *write__add = temp
5    read__add = read__add + added__val;
    write__add = write__add + added__val;
    counter = counter + added__val;
}

```

が表示される。このような代入・加算等の処理を示す「do」が記述されている場合は、ステップC 8に移行して現在のプログラムアドレスを記憶する。その後、ステップC 14に移行する。

ステップC 7で「do」が記述されていない場合はステップC 9に移行してC言語のプログラムでその間の処理を示す「while」が記述されているかをチェックする。このとき、表示装置24のプログラム記述画面P 1には、例えば、「while」を示す

```
while (counter !=end__val) ;
```

が表示される。このような比較・分岐等の処理を示す「while」が記述されている場合は、ステップC 10に移行して代入・演算処理を実行する。例えば、図21に示すサブルーチンをコールして、そのフローチャートのステップE 1でC言語のプログラムにおいて、当該行が「while」を記述している行かがチェックされる。

「while」が記述されている行の場合は、ステップE 2に移行して後続の処理で生成する命令をcmp命令とする。その後、ステップE 6に移行する。

ステップE 1で「while」が記述されていない行の場合は、ステップE 3に移行して演算処理は加算かをチェックする。演算処理が加算の場合はステップE 4に移行して後続の処理で生成する命令をadd命令とする。演算処理が加算ではない場合はステップE 5に移行して後続の処理で生成する命令をload命令とする。その後、ステップE 6に移行する。ステップE 6ではレジスタr iに書き込まれる変数に対応するレジスタ番号と、レジスタアレイ11'から読み出される変数に対応するレジスタr iのレジスタ番号を調べられる。書込みアドレスAw

及び読み出しアドレス A_r を決めるためである。その後、ステップ E 7 に移行する。

- ステップ E 7 では両方のレジスタ番号が「32」以上かをチェックする。両方のレジスタ番号が「32」以上の場合はステップ E 8 に移行して図 17 A に示した命令形態 # F 1 で命令を生成する。この命令形態 # F 1 で「レジスタ種類 1」及び「レジスタ種類 2」には「1」が記述される。このとき、「レジスタ種類 1」及び「レジスタ種類 2」は圧縮プログラムの命令構造の中に記述される。例えば、第 32 番から第 8191 番のグループのレジスタ $r_{32} \sim r_{8191}$ に関して「レジスタ種類 1」及び「レジスタ種類 2」に「1」が記述される。その後、
- 10 図 20 に示したメインルーチンのステップ C 10 にリターンする。

- また、ステップ E 7 で両方のレジスタ番号が「32」以上ではない場合はステップ E 9 に移行して両方のレジスタ番号が「31」以下かをチェックする。ここで両方のレジスタ番号が「31」以下の場合はステップ E 10 に移行して図 17 D に示した命令形態 # F 4 で命令を生成する。この命令形態 # F 4 で「レジスタ種類 1」及び「レジスタ種類 2」には「0」が記述される。このとき、「レジスタ種類 1」及び「レジスタ種類 2」は圧縮プログラムの命令構造の中に記述される。例えば、第 0 番から第 31 番のグループのレジスタ $r_0 \sim r_{31}$ に関して「レジスタ種類 1」及び「レジスタ種類 2」に「0」が記述される。その後、図 20 に示したメインルーチンのステップ C 10 にリターンする。
- 15

- 更に、両方のレジスタ番号が「31」以下ではない場合はステップ E 11 に移行してレジスタアレイ 11' から読み出される変数のレジスタ r_i の番号が「32」以上かをチェックする。読み出される変数のレジスタ r_i の番号が「32」以上の場合は、ステップ E 12 に移行して図 17 B に示した命令形態 # F 2 で命令を生成する。この命令形態 # F 2 で「レジスタ種類 1」に「0」が記述され、
- 20 「レジスタ種類 2」には「1」が記述される。その後、図 20 に示したメインルーチンのステップ C 10 にリターンする。

更にまた、レジスタアレイ 11' から読み出される変数のレジスタ r_i の番号が「32」以上ではない場合は、ステップ E 13 に移行して図 17 C に示した命令形態 # F 3 で命令を生成する。この命令形態 # F 3 では「レジスタ種類 1」に

「1」が記述され、「レジスタ種類2」に「0」が記述される。その後、図20
に示したメインルーチンのステップC10にリターンする。その後、ステップC
11に移行してjump命令を生成する。jump命令の飛び先は先に記憶した
プログラムアドレスを用いる。その後、ステップC14に移行する。上述のステ
5 ップC9で「while」が記述されていない場合はステップC12に移行してC言語
のプログラムにおいて、データの代入又は加算かをチェックする。データの代入
又は加算の場合はステップC13に移行してデータの代入又は演算処理を実行す
る。このステップC13では、図21に示したサブルーチンをコールして、その
フローチャートのステップE1～E13を経て図20に示したメインルーチンの
10 ステップC13にリターンする。その後、ステップC14に移行する。

また、ステップC12でC言語のプログラムにおいて、データの代入又は加算
ではない場合はステップC14に移行する。ステップC14ではC言語のプログ
ラムに関して最後の行かをチェックされる。最後の行ではない場合は、ステッ
C15に移行してプログラムアドレスを進める。その後、ステップC2に戻って
15 上述したコンパイル処理を繰り返すようになされる。最後の行に至ってこのコン
パイル処理を終了する。これにより、図17A～図17Eに示したような命令形
態#F1～#F5であって、命令長の異なる圧縮プログラムAPを作成することが
できる。この圧縮プログラムAPにおいて、第0番から第31番のグループの
レジスタr0～r31を指定する命令ビット数に関してはm=5ビットであり、
20 第32番から第8191番のグループのレジスタr32～r8191を指定する
命令ビット数に関してはn=13ビットである。

続いて、プログラム実行系IIにおける処理例について説明をする。図22は復
元された演算プログラムによる演算命令の例を示す表図である。図23はレジス
タr0, r1・・・r32, r33, r34, r35等の状態例、図24は外
25 部メモリ2におけるデータ格納例を各々示す図である。この例では、演算処理装
置100'に接続された外部メモリ2の中に図24に示すような10個のメモリ
セルの配列を二組用意する。一方はメモリ配列#M1で、他方はメモリ配列#M
2である。そして、図22に示す8つの演算命令(Instruction)#I1～#I
8に基づいて、その一組のメモリ配列#M1に格納された値に「1」を加算し、

もう一組のメモリ配列#M2にその結果を格納する演算処理の例を挙げる。

図22に示す演算命令(Instruction) #I1~#I8はROM14の圧縮プログラムAPを復元した後の演算プログラムによるものである。この演算プログラムでは図23に示すように、アクセス頻度が高いレジスタriが2つあるため、
5 これらをそれぞれr0とr1に割り当てた。これにより、プログラム全体の長さを圧縮する前の演算プログラムに比べて効率よく短縮することができた。また、図23において、レジスタ番号r0で示されるレジスタは一時的に使用され、レジスタ番号r1で示されるレジスタには加算値「1」が格納される。また、レジスタ番号r32で示されるレジスタには読み出しアドレス「0」が格納され、
10 レジスタ番号r33で示されるレジスタには書込みアドレス「10」が格納され、レジスタ番号r34で示されるレジスタにはカウンタの初期値「0」が格納され、レジスタ番号r35で示されるレジスタには演算回数値(終了値)「10」が格納される。

図22に示す各々の演算命令#I1~#I8には、ニーモニックによる表現、
15 機械語による表現及び処理の内容が示されている。演算命令#I1は図22に示した命令構造において、機械語で140020hによって表されるload, r0, (r32)であり、レジスタアレイ11'のレジスタr32の値をアドレスとし、外部メモリ2から読み出した値をレジスタr0に格納する内容である。動作としては例えば、レジスタr32の値を「0」としたとき、図24に示した外部
20 部メモリ2の読み出しアドレスが「0」の内容である、メモリ配列#M1のデータ「0」が読み出され、このデータ「0」がレジスタr0に書き込まれる。

演算命令#I2は機械語で4001hによって表されるadd, r0, r1であり、レジスタアレイ11'のレジスタr0にレジスタr1の値を加算し、その結果をレジスタr0に格納する内容である。動作としてはレジスタr0の内容である「0」にレジスタr1の値である「1」が加算され、その結果「1」がレジスタr0に書き込まれる。演算命令#I3は機械語で280420hによって表されるload, (r33), r0であり、レジスタアレイ11'のレジスタr33の値をアドレスとして、レジスタr0の値を外部メモリ2に書き込む内容である。動作としてはレジスタr33が示す外部メモリ2のメモリ配列#M2のア

ドレスにデータ「1」が書き込まれる。

演算命令# I 4は機械語で480401hによって表されるadd, r32, r1であり、レジスタアレイ11'のレジスタr32にレジスタr1の値を加算して、その結果をレジスタr32に格納する内容である。動作としてはレジスタ
5 r32の内容である「0」にレジスタr1の値である「1」が加算され、その結果「1」がレジスタr32に書き込まれる。演算命令# I 5は機械語で480421hによって表されるadd, r33, r1であり、レジスタアレイ11'のレジスタr33にレジスタr1の値を加算して、その結果をレジスタr33に格納する内容である。動作としてはレジスタr33の内容である「0」にレジスタ
10 r1の値である「1」が加算され、その結果「1」がレジスタr33に書き込まれる。

演算命令# I 6は機械語で480441hによって表されるadd, r34, r1であり、レジスタアレイ11'のレジスタr34にレジスタr1の値を加算して、その結果をレジスタr34に格納する内容である。動作としてはレジスタ
15 r34の内容である「0」にレジスタr1の値である「1」が加算され、その結果「1」がレジスタr34に書き込まれる。この演算命令# I 4～# I 6によって実行ステートマシン51内のカウンタでは読み出しアドレスAr及び書き込みアドレスAwに関して「1」が加算される。演算命令# I 7は機械語で8C044023hによって表されるcmp, r34, r35であり、レジスタアレイ
20 11'のレジスタr34の内容とレジスタr35の内容とを比較し、その値が同じ場合は、zero flagに「1」をセットし、異なっている場合は「0」にセットする内容である。動作としてはレジスタr34とレジスタr35の値である「1」と「10」は異なるので、zero flagには「0」がセットされる。zero flagの値はラッチ回路510によって保持され、以降の命令によって参照される。

25 演算命令# I 8は機械語でE00000hによって表されるjump nz, LOOPであり、zero flagが「0」の場合は、LOOPで示されるラベルへ制御を移す内容である。動作としては、zero flagが「0」の場合は制御を演算命令# I 1に移す。上記の動作が10回、繰り返されるとレジスタr34の値が「10」になり、演算命令# I 7によりzero flagが「1」にセットされ、演算命令# I 8で

制御が移らなくなり、演算処理を終了するようになされる。このように、全てのレジスタ r_i の命令ビット数を単一の方法で表現した場合と比べて、効率良く ROM 14 を使用することが可能になる。

続いて、マイクロプロセッサ 101 における動作例について説明をする。図 25 はマイクロプロセッサ 101 における動作例を示すフローチャートである。図 26 A 及び図 26 B は命令ビット復元デコーダ 13 における処理例を示すフローチャートである。

この実施例ではマイクロプロセッサ 101 がプログラム実行系 II を構成し、ROM 14 から読み出された圧縮プログラム AP から図 22 に示した演算命令 # I 1 ~ # I 8 を含む演算プログラムを復元する。このとき、命令形態 # F 1 ~ # F 4 に関して「レジスタ種類 1」及び、「レジスタ種類 2」にコード「0」が記述されている場合は、命令ビット復元デコーダ 13 によってレジスタ番号の拡張が行われる。この際の命令ビットの拡張では、例えば、「レジスタ番号 1」の命令ビット数 $m=5$ ビットの上位に 8 ビットの「0」を追加するようになされる。この演算プログラムに基づいて、図 24 に示した外部メモリ 2 の中のメモリ配列 # M1 の値に「1」を加算し、メモリ配列 # M2 に格納するようになされる。

レジスタアレイ 11' のレジスタ状態については、図 23 に示したように例えば、6 個のレジスタ r_0 , r_1 , r_{32} , r_{33} , r_{34} , r_{35} に関して、 r_0 が不定、 r_1 が初期値「1」、 r_{32} 及び r_{34} が共に初期値「0」、 r_{33} 及び r_{35} が初期値「10」が設定される。これらの値を書き込む場合は、実行ステートマシーン 51 ではレジスタアレイ 11' に書き込みアドレス A_w が出力され、その初期値「0」、「1」、「10」が設定される。これを動作条件にして、図 25 に示すフローチャートのステップ F1 で、まず、命令ビット復元デコーダ 13 は ROM 14 から圧縮プログラム（機械語命令）AP を順次受け取り、このプログラム AP を解読して所定の命令長の演算命令 # I 1 ~ # I 8 を検出する。

この命令ビット復元デコーダ 13 は例えば、図 26 A に示すサブルーチンをコールしてそのフローチャートのステップ G1 で命令部分を取り出し、命令信号 S9 を実行ステートマシーン 51 に出力する。これと共に、命令ビット復元デコー

ダ 1 3 ではステップ G 2 に移行して当該命令形態が j u m p 命令であることをチェックする。当該命令形態が # F 5 で示される j u m p 命令の場合はステップ G 1 2 に移行して flag condition、jump address を出力する。その後、メインルーチンのステップ F 1 にリターンする。また、ステップ G 2 で当該命令形態が j u m
5 p 命令でない場合は、ステップ G 4 に移行して当該命令形態に関して「レジスタ種類 1」に記述されているコードは「0」又は「1」かをチェックする。「レジスタ種類 1」にコード「0」が記述されている場合はステップ G 4 に移行して
「レジスタ番号 1」の命令ビット数 m を 5 b i t 長として圧縮プログラム A P から取り出す。その後、ステップ G 5 に移行して「レジスタ番号 1」の命令ビット
10 数 m = 5 b i t の上位 8 ビットに「0」を付加して 1 3 b i t 長とする。その後、図 2 6 B に示すフローチャートのステップ G 7 に移行する。

なお、上述のステップ G 3 で「レジスタ種類 1」にコード「1」が記述されている場合はステップ G 6 に移行して「レジスタ番号 1」の命令ビット数 n を 1 3
b i t 長として圧縮プログラム A P から取り出す。そして、図 2 6 B に示すフロー
15 チャートのステップ G 7 に移行して、当該命令形態に関して「レジスタ種類 2」に記述されているコードは「0」又は「1」かをチェックする。「レジスタ種類 2」にコード「0」が記述されている場合はステップ G 8 に移行して「レジスタ番号 2」の命令ビット数 m を 5 b i t 長として圧縮プログラム A P から取り出す。その後、ステップ G 9 に移行して「レジスタ番号 2」の命令ビット数 m =
20 5 b i t の上位 8 ビットに「0」を付加して 1 3 b i t 長とする。その後、ステップ G 1 1 に移行する。

上述のステップ G 7 で「レジスタ種類 2」にコード「1」が記述されている場合はステップ G 1 0 に移行して「レジスタ番号 2」の命令ビット数 n を 1 3 b i
t 長として圧縮プログラム A P から取り出す。その後、ステップ G 1 1 に移行して、
25 「レジスタ番号 1」、「レジスタ番号 2」、「アクセス方法 # 1」及び「アクセス方法 # 2」を検出する。その後、図 2 5 に示したメインルーチンのステップ F 1 にリターンする。従って、実行ステートマシン 5 1 には「レジスタ種類 1」及び「レジスタ種類 2」は出力されず、所定の命令長の演算命令 # I 1 ~ # I 8 に基づく命令制御信号 S 4、命令信号 S 9 及び各引数信号 S 1 0 が出力され

る。

この命令信号S 9にはl o a d命令、a d d命令、c m p命令、j u m p命令が含まれる。各引数信号S 1 0にはアクセス方法# 1, アクセス方法# 2, レジスタ番号r 0, r 1 . . . 等、フラグ状態 (flag condition) 及びジャンプアドレス等が含まれる。命令制御信号S 4はデコーダ1 3から命令読出しステートマシーン5 2に出力される。なお、ROM1 4で圧縮プログラムA Pを読み出す場所 (アドレス) はプログラムカウンタ5 4 (P C) によって指定される。これらの読み出しの動作は命令読出しステートマシーン5 2によって制御される。

命令読出しステートマシーン5 2では命令ビット復元デコーダ1 3から出力される命令制御信号S 4に基づいてプログラムカウンタ5 4及び実行ステートマシーン5 1を制御する。例えば、当該マシーン5 2は命令ビット復元デコーダ1 3から命令信号S 9及び各引数信号S 1 0が実行ステートマシーン5 1へ出力されると共に命令実行開始信号S 2 9を出力する。実行ステートマシーン5 1では命令実行開始信号S 2 9に基づいて命令の実行を開始する。例えば、データの書込み時には、書込み制御信号S wがレジスタアレイ1 1' に出力され、セクタ5 9には選択制御信号S 2 4が出力される。データの読出し時には、レジスタアレイ1 1' に読出しアドレスA rが出力される。

演算時には、ラッチ制御信号S 3 4がラッチ回路5 8に出力され、ラッチ回路5 1 0にはラッチ制御信号S 3 8が出力される。当該プロセッサ外部には外部制御信号S 1 6が出力される。命令の実行が終了すると、実行ステートマシーン5 1は命令読出しステートマシーン5 2へ実行終了信号S 2 6を出力し、プログラムカウンタ5 4の値を進めるようになされる。プログラムカウンタ5 4ではカウント制御信号S 3 0に基づいてROM1 4から圧縮プログラムA Pを読み出す場所が指定される。+ 1インクリメンタ5 5はプログラムカウンタ5 4のカウント出力信号S 5を「+ 1」してインクリメントするようになされる。

そして、ステップF 2で実行ステートマシーン5 1は命令読出しステートマシーン5 2の命令読出し制御を受けて演算命令# I 1を受け取ると、機械語で1 4 0 0 2 0 hによって表されるl o a d, r 0, (r 3 2) に基づいて書込み信号S 1 6を外部メモリ 2に出力する。この値はレジスタアレイ1 1' の読み出しA

ドレスA_rとして用いられる。レジスタアレイ11'は32番目の値をデータ信号線L20に出力する。この値はラッチ回路511によって保持され、アドレスバス19Bを経由し、外部メモリ2へ出力される。そして、レジスタアレイ11'のレジスタr32の値をアドレスとし、外部メモリ2から読み出した値をレジスタr0に格納する。このとき、レジスタr32の値が「0」であるので、図24に示した外部メモリ2のアドレスが「0」の内容であるメモリ配列#M1のデータ「0」が読み出され、このデータ「0」がレジスタr0に書き込まれる。

つまり、外部メモリ2からアドレスバス19Bによって転送されたアドレス(場所)のデータがセクタ59に出力される。実行ステートマシーン51ではこのデータが選択されるように、選択信号S24を出力する。これにより、データがレジスタアレイ11'に入力される。そして、実行ステートマシーン51では書き込みアドレスA_w＝「0」をレジスタアレイ11'に出力する。その後、実行ステートマシーン51は書き込み信号S_wを用いて実際に、演算結果値の書き込みを指示するようになされる。その後、ステップF3で実行ステートマシーン51は命令読出しステートマシーン52の命令読出し制御を受けて演算命令#I2を受け取ると、機械語で4001hによって表されるadd, r0, r1に基づいてレジスタアレイ11'のレジスタr0にレジスタr1の値を加算し、その結果をレジスタr0に格納する。このとき、レジスタr0の内容である「0」にレジスタr1の値である「1」が加算され、その結果「1」がレジスタr0に書き込まれる。

そして、ステップF4で実行ステートマシーン51は命令読出しステートマシーン52の命令読出し制御を受けて演算命令#I3を受け取ると、機械語で280420hによって表されるload, (r33), r0に基づいてレジスタアレイ11'のレジスタr33の値をアドレスとして、レジスタr0の値を外部メモリ2に書き込む。このとき、レジスタr33が示す外部メモリ2のメモリ配列#M2のアドレスにデータ「1」が書き込まれる。その後、ステップF5で実行ステートマシーン51は命令読出しステートマシーン52の命令読出し制御を受けて演算命令#I4を受け取ると、機械語で480401hによって表されるadd, r32, r1に基づいてレジスタアレイ11'のレジスタr32にレジスタr1

の値を加算して、その結果をレジスタ r_{32} に格納する。このとき、レジスタ r_{32} の内容である「0」にレジスタ r_1 の値である「1」が加算され、その結果「1」がレジスタ r_{32} に書き込まれる。この演算命令 # I 4 によって実行ステートマシン 51 内のカウンタでは読み出しアドレス A_r 及び書き込みアドレス A_w に関して「1」が加算される。

そして、ステップ F 6 で実行ステートマシン 51 は命令読出しステートマシン 52 の命令読出し制御を受けて演算命令 # I 5 を受け取ると、機械語で 4 8 0 4 2 1 h によって表される add, r_{33}, r_1 に基づいてレジスタアレイ 11' のレジスタ r_{33} にレジスタ r_1 の値を加算して、その結果をレジスタ r_{33} に格納する。このとき、レジスタ r_{33} の内容である「0」にレジスタ r_1 の値である「1」が加算され、その結果「1」がレジスタ r_{33} に書き込まれる。この演算命令 # I 5 によって実行ステートマシン 51 内のカウンタでは読み出しアドレス A_r 及び書き込みアドレス A_w に関して「1」が加算される。その後、ステップ F 7 で実行ステートマシン 51 は命令読出しステートマシン 52 の命令読出し制御を受けて演算命令 # I 6 を受け取ると、機械語で 4 8 0 4 4 1 h によって表される add, r_{34}, r_1 に基づいてレジスタアレイ 11' のレジスタ r_{34} にレジスタ r_1 の値を加算して、その結果をレジスタ r_{34} に格納する。このとき、レジスタ r_{34} の内容である「0」にレジスタ r_1 の値である「1」が加算され、その結果「1」がレジスタ r_{34} に書き込まれる。この演算命令 # I 6 によって実行ステートマシン 51 内のカウンタでは読み出しアドレス A_r 及び書き込みアドレス A_w に関して「1」が加算される。

この例では、ステップ F 8 でレジスタ r_{35} が示す値 = 10 回に至ったかが判別される。例えば、実行ステートマシン 51 は命令読出しステートマシン 52 の命令読出し制御を受けて演算命令 # I 7 を受け取ると、機械語で 8 C 0 4 4 0 2 3 h によって表される cmp, r_{34}, r_{35} に基づいてレジスタアレイ 11' のレジスタ r_{34} の内容とレジスタ r_{35} の内容とを比較し、その値が同じ場合は、zero flag に「1」をセットし、異なっている場合は「0」にセットする。このとき、レジスタ r_{34} とレジスタ r_{35} の値である「1」と「10」は異なるので、zero flag には「0」がセットされる。zero flag の値はラッチ回路 51

0によって保持され、以降の命令によって参照される。そして、実行ステートマシン51は命令読出しステートマシン52の命令読出し制御を受けて演算命令#18を受け取ると、機械語でE00000hによって表されるjump nz, LOOPに基づいてzero flagが「0」の場合は、LOOPで示されるラベルへ制御を移行する。zero flagが「0」の場合は制御をステップF2の演算命令#11に移す。上記の動作がステップF8で10回繰り返されるとレジスタr34の値が「10」になり、演算命令#17によりzero flagが「1」にセットされ、演算命令#18で制御が移らなくなり、演算処理を終了するようになされる。

このように、本発明に係る第4の実施例としてのマイクロプロセッサ101によれば、8192個のレジスタr0～r8191の中でその使用頻度に基づいて当該レジスタriを指定する命令ビット数が予め圧縮されると共に、当該プログラムの命令構造の中に「レジスタ種類1」及び「レジスタ種類2」が記述された命令長の異なる圧縮プログラムAPに基づいてデータを処理するようになされる。ROM14には8192個のレジスタr0～r8191の中から当該レジスタriを指定するための圧縮プログラムAPが記憶される。命令ビット復元デコーダ13では、このROM14から圧縮プログラムAPを読み出して「レジスタ種類1」及び「レジスタ種類2」が解読され、この「レジスタ種類1」及び「レジスタ種類2」に基づいて当該レジスタriを指定するための命令ビット数が復元される。

従って、レジスタriの使用頻度に応じて可変された命令の長さの圧縮プログラムAPであって、頻繁にアクセスするレジスタr0～r31には短い長さの命令がセットされた、圧縮プログラムデータをROM14に格納することができるので、そのメモリ容量を低減することができる。この例ではプログラムを圧縮する前に比べて16ビット×32個×命令数分だけメモリ容量を低減できる。これにより、メモリセルや論理演算素子から成るPLDによりマイクロプロセッサ101を構築する場合に、ROMとして機能させるメモリセルの占有率を低減することができ、その分のメモリセルをレジスタに多く割り当てることができるようになる。

この実施例では演算処理装置100'に外部メモリ2を接続してマイクロプロ

- セッサ 101 を構成し、複数のレジスタを指定してレジスタ相対メモリアドレッシング処理を実行する演算プログラムに基づいて演算処理を実行する場合について説明したが、これに限られることはなく、第 1 の実施例と同様にして、レジスタ相対レジスタアドレッシング処理を実行するための演算命令を圧縮プログラム
- 5 AP に基づいて演算処理を実行するようにしてもよい。この場合には、図 15 に示したレジスタアレイ 11' の入力部分に、図 1 に示したようなセクタ 56, 57 を接続し、このセクタ 56, 57 を実行ステートマシン 51 で制御すればよい。このようにすると、第 1 の実施例のレジスタアレイ 11 を随時書込み及び読出し可能なメモリのように動作できるという効果、基板占有面積を低減できる
- 10 という効果に加えて、第 2 の実施例のプログラムデータを格納する ROM 等のメモリ容量を低減できるという効果とを重複して得られるようになる。

産業上の利用可能性

- この発明は各種電子機器に内蔵可能でプログラム可能な 1 チップマイクロコンピュータ等に、また、命令実行プログラム等に基づいて各種データ処理をする CPU や、MPU、PLD 等、これらの組み込み電子機器に適用して極めて好適である。
- 15

請 求 の 範 囲

1. 任意の演算プログラムに基づいて演算処理をする装置であって、
書込みアドレス及び書込み制御信号に基づいて任意の値を保持し、及び、読出
5 しアドレスに基づいて当該値を出力するレジスタを複数有したレジスタアレイと、
前記レジスタアレイから読み出された値を演算する演算部と、
前記演算部を動作させるための演算プログラムから演算命令を解読する命令解
読部と、
前記命令解読部によって解読された演算命令を実行するために前記レジスタア
10 レイ及び演算部を制御する命令実行制御部とを備え、
前記命令実行制御部は、
前記演算命令に基づいて一の前記レジスタを選択し、
選択された前記レジスタが保持する値によって他のレジスタを選択するレジス
タ相対レジスタアドレッシング処理を実行することを特徴とする演算処理装置。
15
2. 前記演算プログラムを格納した読出し専用メモリセルを備えることを特徴と
する請求項 1 に記載の演算処理装置。
3. 前記演算プログラムには、
20 前記レジスタ相対レジスタアドレッシング処理を実行するための演算命令を含
むことを特徴とする請求項 1 に記載の演算処理装置。
4. 前記レジスタアレイ及び読出し専用メモリが複数のメモリセルにより構成さ
れ、
25 前記演算部、命令解読部及び命令実行制御部が複数の算術論理演算素子により
構成され、
前記メモリセル及び算術論理演算素子を同一半導体チップ上に備えたプログラ
マブル・ロジック・デバイスによって構成されることを特徴とする請求項 1 に記
載の演算処理装置。

5. 前記命令実行制御部には、

前記一のレジスタを選択するための読出し実行アドレス又は当該レジスタを再度選択するための読出しアドレスのいずれか一方を選択する第1のセクタと、

- 5 前記一のレジスタを選択するための書込み実行アドレス又は当該レジスタを再度選択するための書込みアドレスのいずれか一方を選択する第2のセクタとを有することを特徴とする請求項1に記載の演算処理装置。

6. 任意の演算プログラムに基づいて演算処理をする装置を構築する方法であって、

- 10 予め同一半導体チップ上に複数のメモリセル及び算術論理演算素子を形成し、
前記メモリセルを組み合わせてレジスタアレイ及び読出し専用メモリを画定すると共に、前記算術論理演算素子を組み合わせて演算部、命令解読部及び命令実行制御部を画定し、その後、前記レジスタアレイ、読出し専用メモリ、演算部、
15 命令解読部及び命令実行制御部を予め設定された配線情報に基づいて結線すると共に、前記読出し専用メモリに任意の演算プログラムを書込むことを特徴とする演算処理装置の構築方法。

7. 前記演算プログラムには、

- 20 前記レジスタ相対レジスタアドレッシング処理を実行するための演算命令を含むことを特徴とする請求項6に記載の演算処理装置の構築方法。

8. 前記配線情報を格納する書き換え可能な不揮発性の記憶装置が設けられ、電源オンと共に前記記憶装置から配線情報を読み出して設定し、

- 25 設定された前記配線情報に基づいて前記レジスタアレイ、読出し専用メモリ、演算部、命令解読部及び命令実行制御部間を結線することを特徴とする請求項6に記載の演算処理装置の構築方法。

9. 前記記憶装置に格納される配線情報は当該演算処理装置の機能に応じて随時

書き換えられることを特徴とする請求項6に記載の演算処理装置の構築方法。

10. 前記レジスタアレイは、

- 書き込みアドレス及び書き込み制御信号に基づいて任意の値を保持し、及び、読出しアドレスに基づいて当該値を出力する複数のレジスタを有することを特徴とする請求項6に記載の演算処理装置の構築方法。

11. 前記命令実行制御部は、

- 前記演算命令に基づいて一の前記レジスタを選択し、
10 選択された前記レジスタが保持する値によって他のレジスタを選択するレジスタ相対レジスタアドレッシング処理を実行することを特徴とする請求項6に記載の演算処理装置の構築方法。

12. 前記命令実行制御部には、

- 15 前記一のレジスタを選択するための読出し実行アドレス又は当該レジスタを再度選択するための読出しアドレスのいずれか一方を選択する第1のセレクトと、
前記一のレジスタを選択するための書き込み実行アドレス又は当該レジスタを再度選択するための書き込みアドレスのいずれか一方を選択する第2のセレクトとが
備えられることを特徴とする請求項6に記載の演算処理装置の構築方法。

20

13. 前記演算部では前記レジスタアレイから読み出された値が演算され、

前記命令解読部では前記演算部を動作させるための演算プログラムから演算命令が解読され、

- 前記命令実行制御部では前記命令解読部によって解読された演算命令を実行するために前記レジスタアレイ及び演算部が制御されることを特徴とする請求項6に記載の演算処理装置の構築方法。

14. 任意の演算プログラムに基づいて演算処理をする方法であって、

書き込みアドレス及び書き込み制御信号に基づいて任意の値を保持し、及び、読出

- シアドレスに基づいて当該値を出力するレジスタを予め複数準備し、その後、
前記演算プログラムから演算命令を解読し、
前記演算命令に基づいて一の前記レジスタを選択し、
選択された前記レジスタが保持する値によって他のレジスタを選択するレジスタ
5 相対レジスタアドレッシング処理を実行すると共に、前記演算命令に基づいて
他の前記レジスタを選択し、
選択された他の前記レジスタが保持する値と前記レジスタ相対レジスタアドレ
ッシング処理によって選択されたレジスタの値とを演算することを特徴とする演
算処理方法。
- 10
- 15 15. 前記レジスタが保持する値によって選択されているレジスタに前記演算の
結果を格納することを特徴とする請求項14に記載の演算処理方法。
16. 前記演算プログラムには、
- 15 前記レジスタ相対レジスタアドレッシング処理を実行するための演算命令を含
むことを特徴とする請求項14に記載の演算処理方法。
17. レジスタ相対レジスタアドレッシング処理を実行する演算プログラムに基
づいて演算処理をする装置であって、
- 20 複数のレジスタと、
前記レジスタを使用する頻度に基づいて、当該レジスタを指定する命令ビット
数が予め圧縮されると共に、前記プログラムの命令構造の中にレジスタ種類が記
述された命令長の異なる圧縮プログラムを記憶する記憶部と、
前記記憶部から圧縮プログラムを読み出してレジスタ種類を解読し、当該レジ
25 スタ種類に基づいて前記レジスタを指定するための命令ビット数を復元する命令
解読部と、
前記命令解読部によって復元された所定長さの命令に基づいて前記レジスタを
指定して任意の演算を実行する命令実行演算部とを備えることを特徴とする演算
処理装置。

18. 前記レジスタ種類は、

N個のレジスタを使用する場合であって、

前記N個のレジスタに第1番から第N番のシリアル番号を付与したとき、

- 5 前記第1番から第K番のグループのレジスタを使用頻度が高い部類に、前記第K+1番から第N番のグループのレジスタを使用頻度が低い部類に分けられることを特徴とする請求項17に記載の演算処理装置。

19. レジスタ相対レジスタアドレッシング処理を実行する演算プログラムに基

- 10 づいて演算処理をする演算方法であって、

前記レジスタを使用する頻度に基づいて、当該レジスタを指定する命令ビット数が予め圧縮される共に、前記プログラムの命令構造の中にレジスタ種類が記述された命令長の異なる圧縮プログラムを記憶し、

前記圧縮プログラムを読み出してレジスタ種類を解読し、

- 15 当該レジスタ種類に基づいて前記レジスタを指定するための命令ビット数を復元し、

復元された前記所定長さの命令に基づいて前記レジスタを指定して任意の演算を実行することを特徴とする演算処理方法。

- 20 20. 前記レジスタ種類は、

N個のレジスタを使用する場合であって、

前記N個のレジスタに第1番から第N番のシリアル番号を付与したとき、

前記第1番から第K番のグループのレジスタを使用頻度が高い部類に、

前記第K+1番から第N番のグループのレジスタを使用頻度が低い部類に分け

- 25 られることを特徴とする請求項19に記載の演算処理方法。

21. 一方で、所定のプログラム言語に基づいて、レジスタ相対レジスタアドレッシング演算を実行するための命令を編集してプログラムを作成するプログラム作成装置と、他方で、当該プログラムと複数のレジスタとを使用して前記レジ

タ相対レジスタアドレッシング演算を実行する演算処理装置とを有する演算処理システムであって、

前記プログラム作成装置は、

- 5 前記レジスタを使用する頻度に基づいて当該レジスタを指定する命令ビット数を圧縮すると共に、当該プログラムの命令構造の中にレジスタ種類を記述して命令長の異なる圧縮プログラムを作成し、

前記演算処理装置は、

前記プログラム作成装置で作成された圧縮プログラムを取得してレジスタ種類を解読し、

- 10 前記レジスタ種類に基づいて当該レジスタを指定する命令ビット数を復元し、所定長さの命令に基づいて複数のレジスタを指定して任意の演算を実行することを特徴とする演算処理システム。

22. 前記演算処理は、

- 15 複数のレジスタと、

前記レジスタを指定するための圧縮プログラムを記憶する記憶部と、

前記記憶部から圧縮プログラムを読み出してレジスタ種類を解読し、当該レジスタ種類に基づいて前記レジスタを指定するための命令ビット数を復元する命令解読部と、

- 20 前記命令解読部によって復元された所定の長さの命令に基づいて前記レジスタを指定して任意の演算を実行する命令実行演算部とを有することを特徴とする請求項21に記載の演算処理システム。

23. 前記レジスタ種類は、

- 25 N個のレジスタを使用する場合であって、

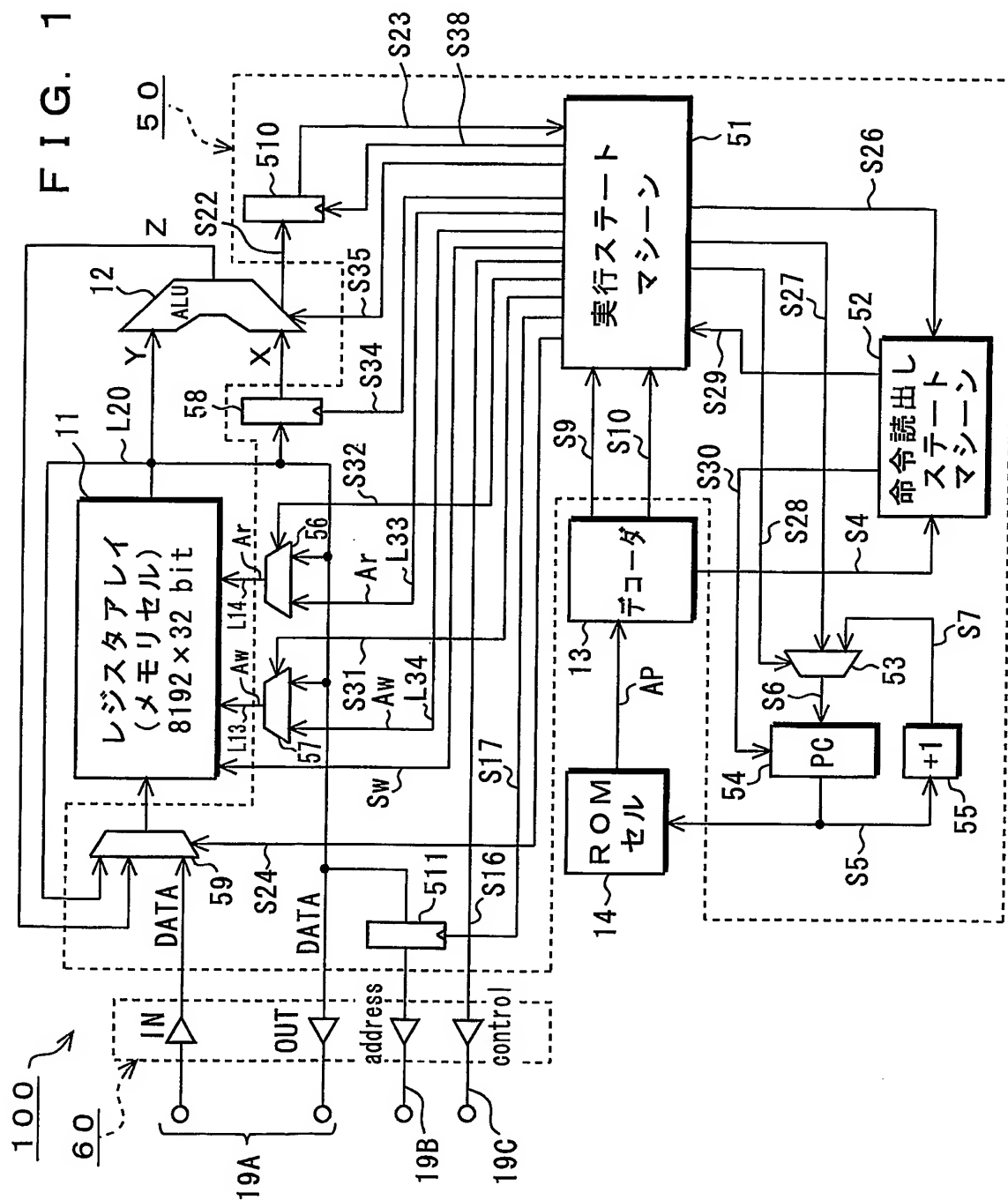
前記N個のレジスタに第1番から第N番のシリアル番号を付与したとき、

前記第1番から第K番のグループのレジスタを使用頻度が高い部類に、

前記第K+1番から第N番のグループのレジスタを使用頻度が低い部類に分けることを特徴とする請求項21に記載の演算処理システム。

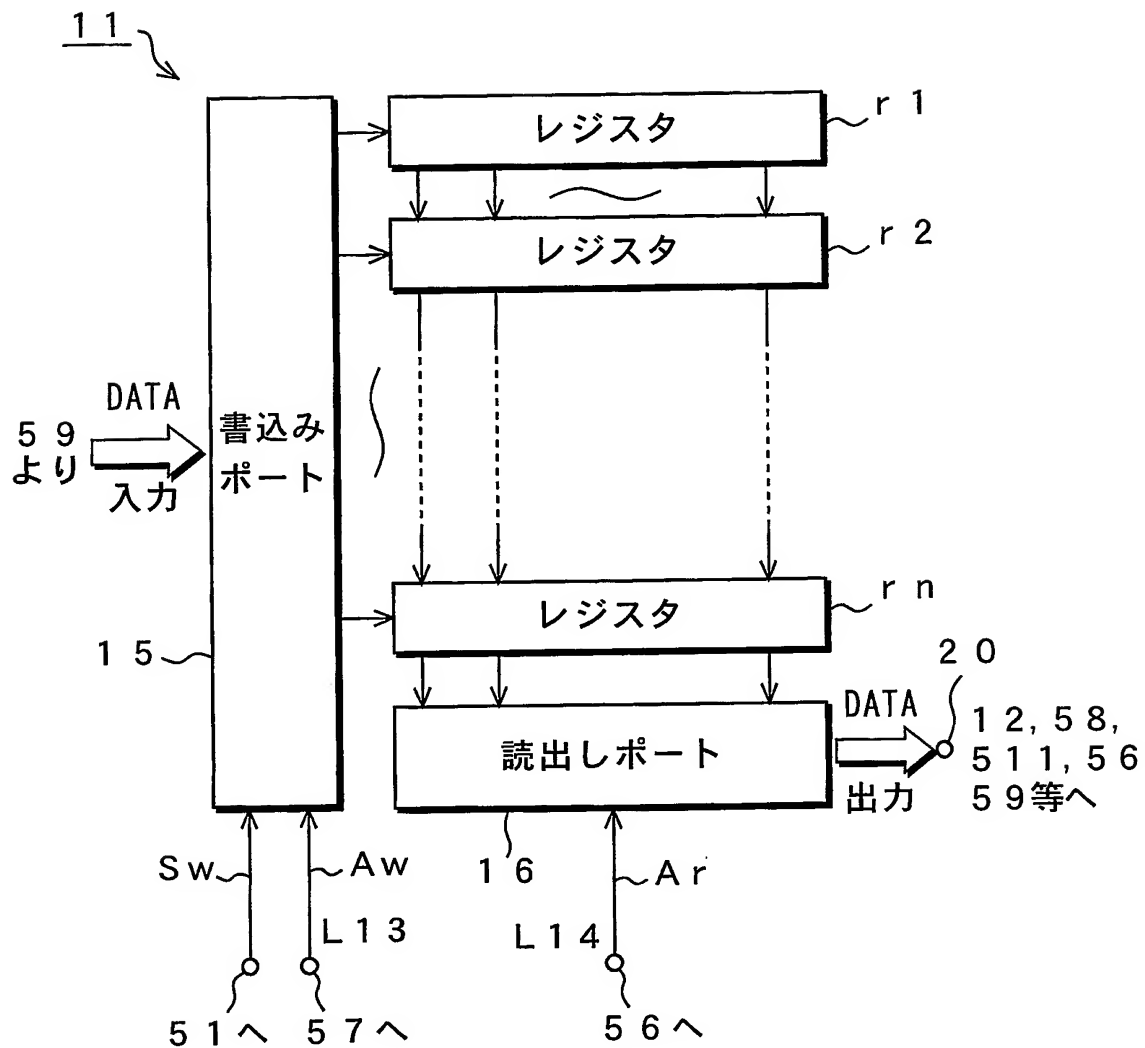
24. プログラム作成系で所定のプログラム言語に基づいてレジスタ相対レジスタアドレッシング演算を実行するための命令を編集してプログラムを作成し、プログラム実行系で当該プログラムと複数のレジスタとを使用して前記レジスタ相対レジスタアドレッシング演算を実行する演算処理方法であって、
- 5 前記プログラム作成系では、
- 前記レジスタを使用する頻度に基づいて当該レジスタを指定する命令ビット数を圧縮すると共に、当該プログラムの命令構造の中にレジスタ種類を記述して命令長の異なる圧縮プログラムを作成し、
- 10 前記プログラム実行系では、
- 前記プログラム作成系で作成された圧縮プログラムを取得してレジスタ種類を解読し、
- 解読された前記レジスタ種類に基づいて当該レジスタを指定する命令ビット数を復元し、
- 15 復元された所定長さの前記命令に基づいて複数のレジスタを指定して任意の演算を実行することを特徴とする演算処理方法。

25. 前記レジスタ種類は、
- N個のレジスタを使用する場合であって、
- 20 前記N個のレジスタ第1番から第N番のシリアル番号を付与したとき、
- 前記第1番から第K番のグループのレジスタを使用頻度が高い部類に、
- 前記第K+1番から第N番のグループのレジスタを使用頻度が低い部類に分けることを特徴とする請求項24に記載の演算処理方法。



2 / 2 5

F I G . 2



3 / 2 5

FIG. 3A

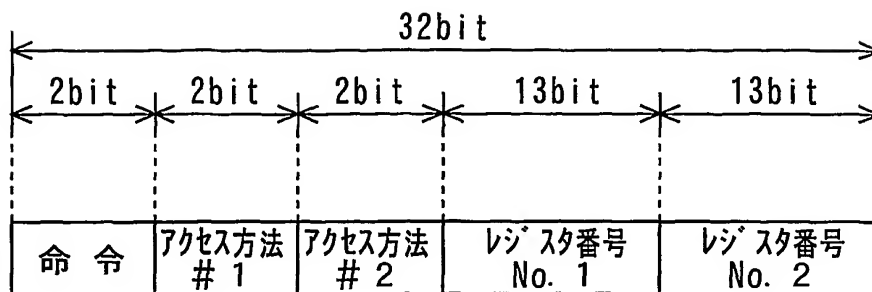
load, add,
cmp 命令

FIG. 3B

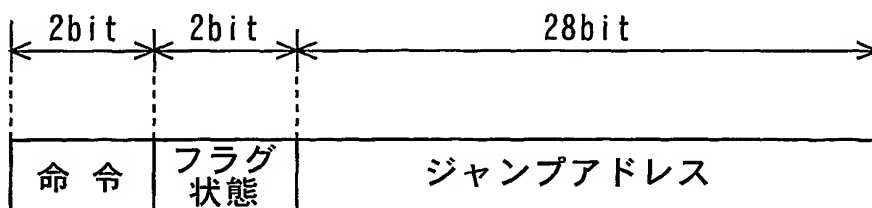
jump
命令

FIG. 3C

命 令	
0	load
1	add
2	cmp
3	jump

FIG. 3D

アクセス方法	
0	レジスタ直接
1	レジスタ相対レジスタ
2	レジスタ相対外部
3	未 使 用

FIG. 3E

フラグ状態	
0	無 条 件
1	zero flag
2	non-zero flag
3	未 使 用

4 / 2 5

F I G . 4

N o .	ニーモニックによる表現		機械語による表現
# 1 1	LOOP:	add[r10], r11	5001400Bh
# 1 2		add r10, r11	4001400Bh
# 1 3		cmp r10, r12	8001400Ch
# 1 4		jump nz, LOOP	E0000000h

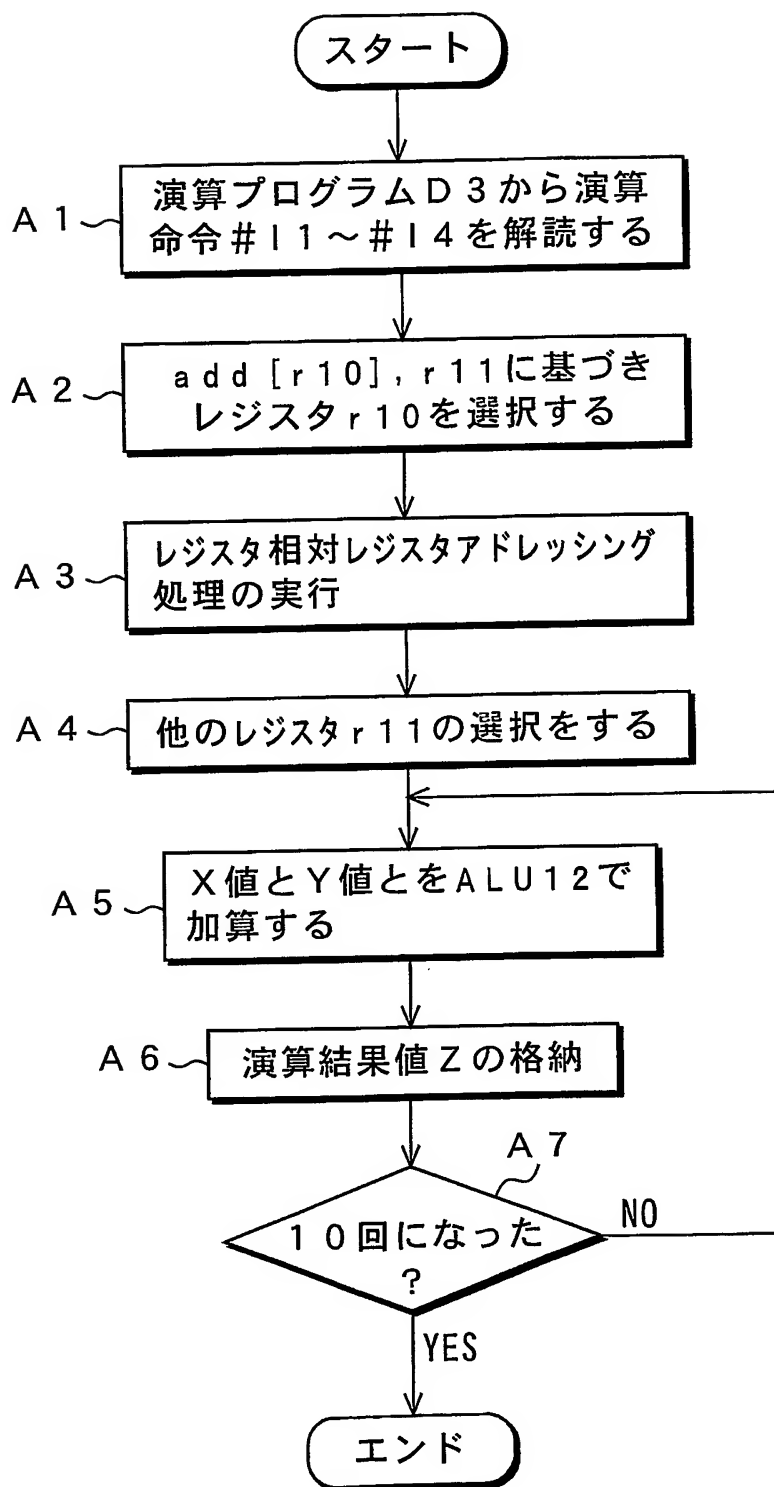
F I G . 5

レジスタ番号		値	
r 0		0	
r 1		0	
r 2		0	
r 3		0	
r 4		0	
r 5		0	
r 6		0	
r 7		0	
r 8		0	
r 9		0	
r 1 0		0	← 現在の配列の番号
r 1 1		1	← 足す値
r 1 2		1 0	← 終了値

1 1 ~

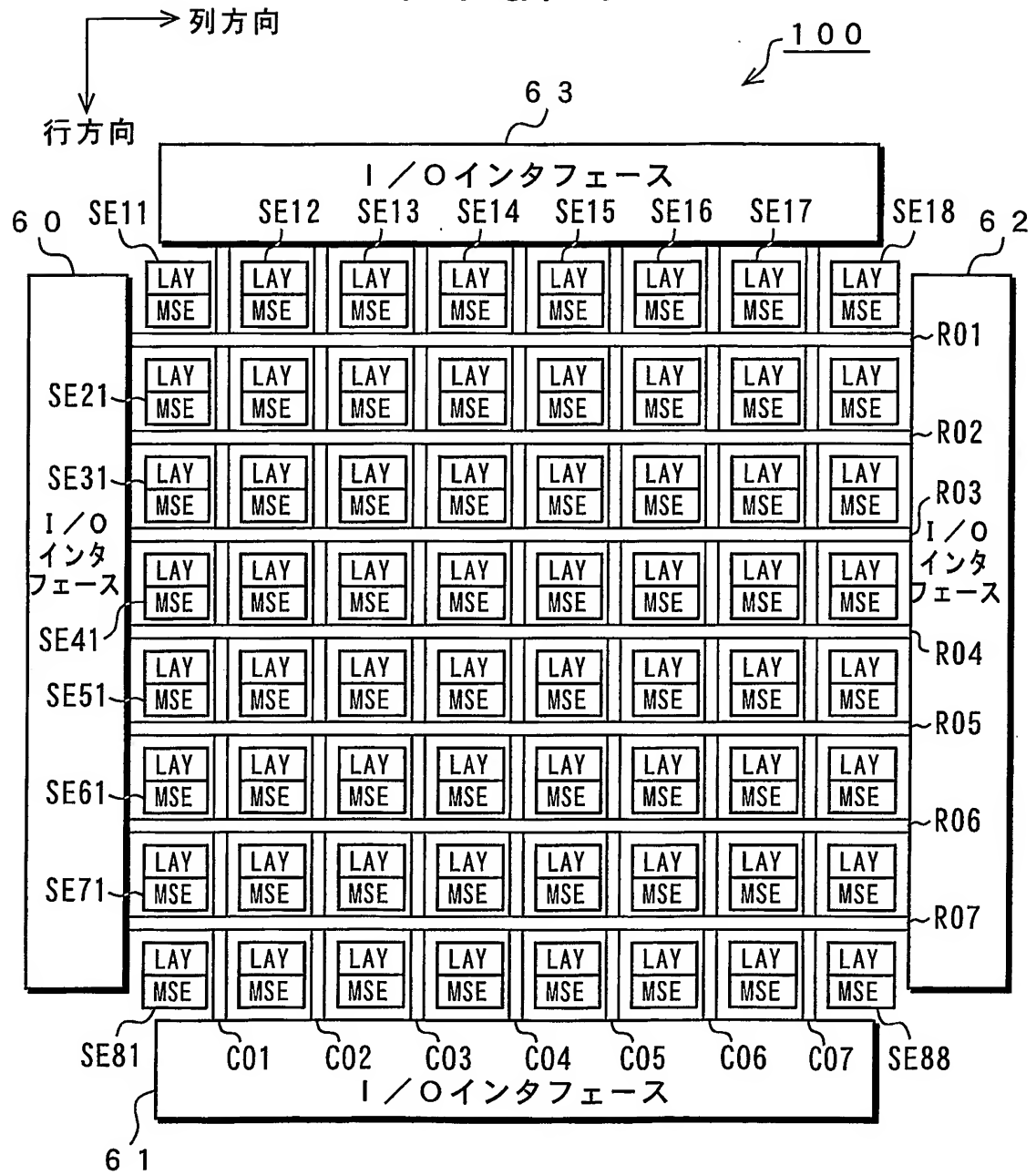
5 / 2 5

F I G . 6



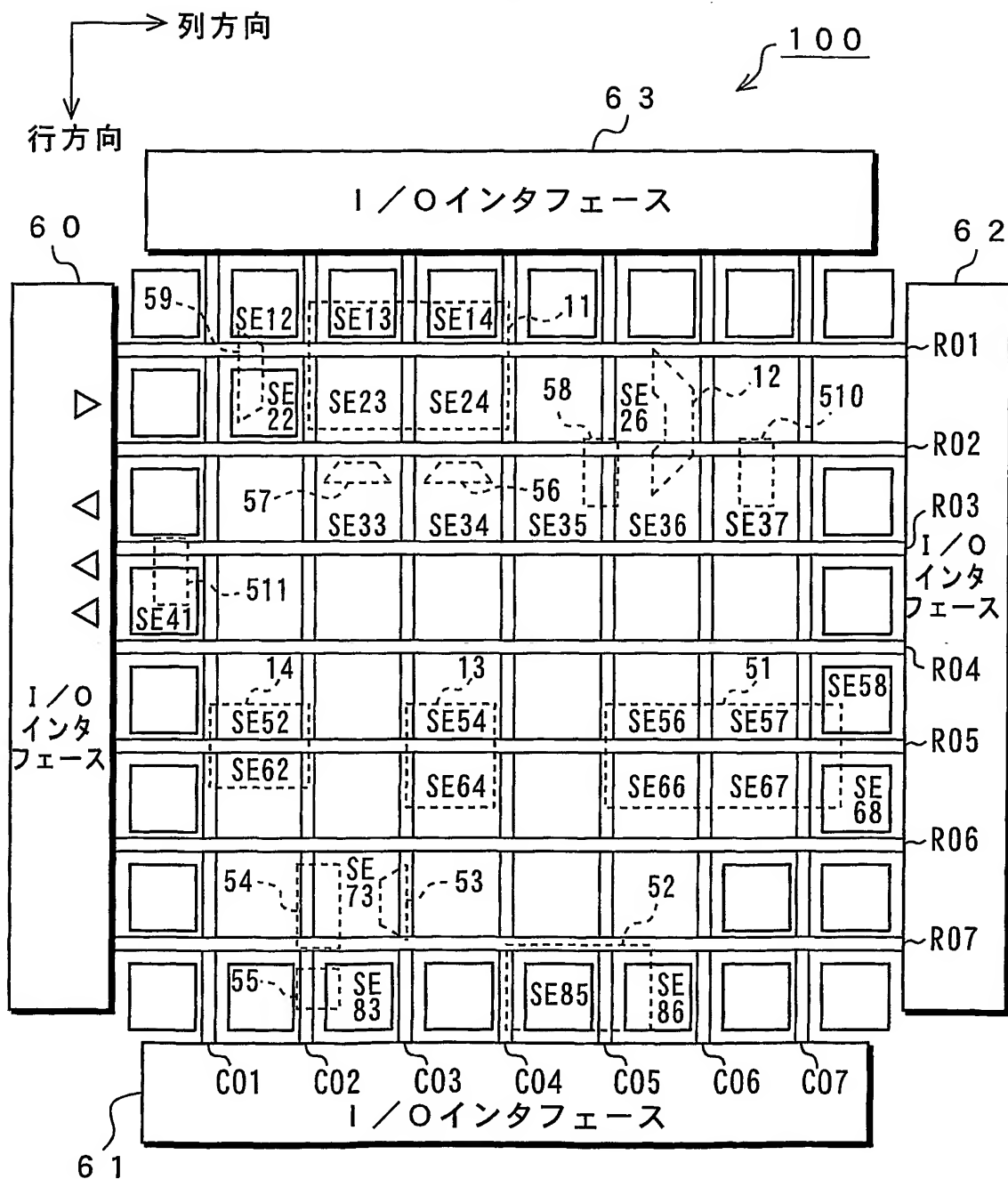
6 / 2 5

FIG. 7



7 / 2 5

F I G . 8



8 / 2 5

FIG. 9

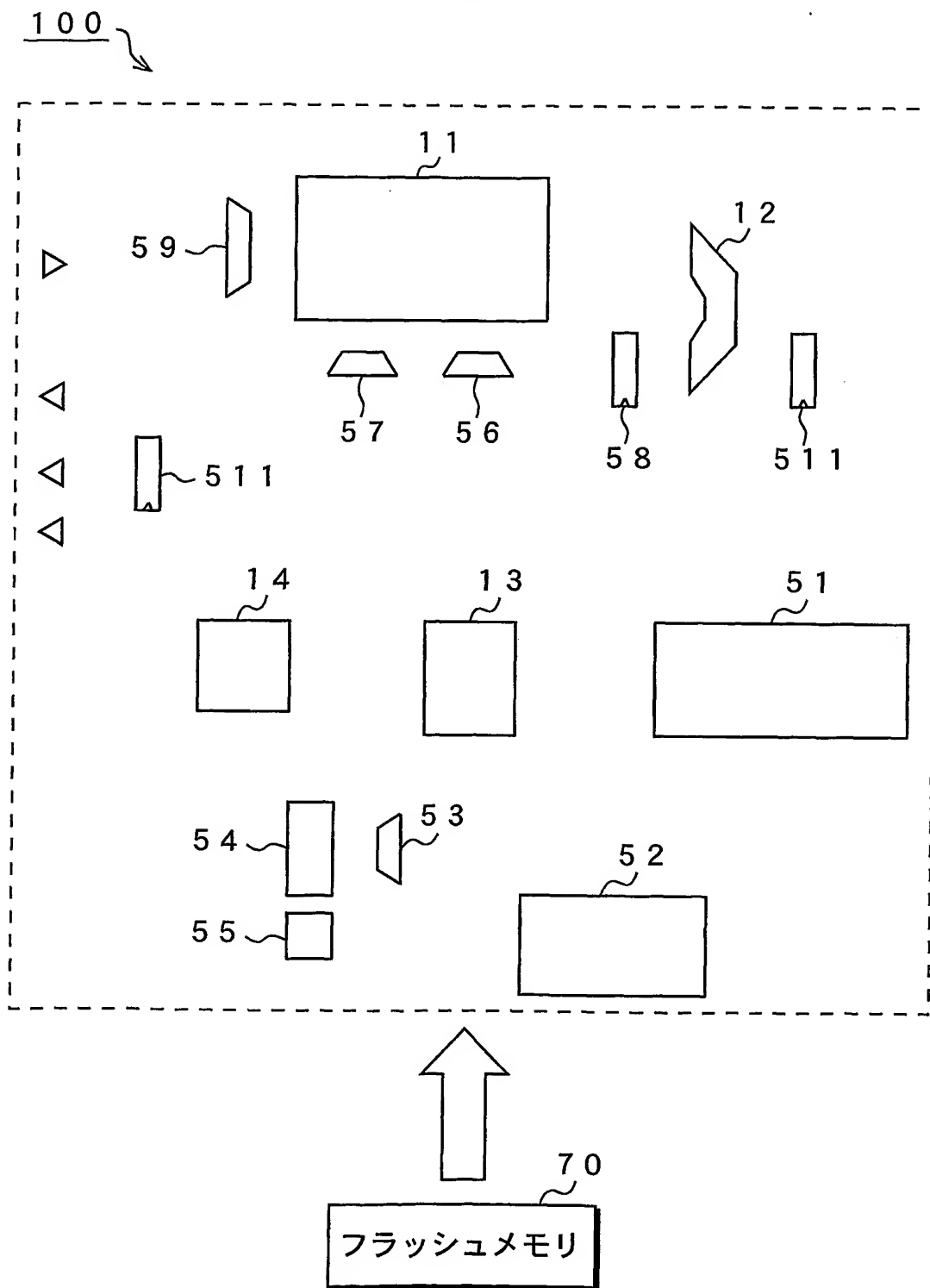


FIG. 10

70
↘

配線情報	信号線	用途	配線情報	信号線	用途
D 1	アキ		D 2 1	L 2 1	演算結果値 Z
D 2	アキ		D 2 2	L 2 2	一致検出信号 S 2 2
D 3	L 3	演算プログラム AP	D 2 3	L 2 3	フラグ状態信号 S 2 3
D 4	L 4	命令制御信号 S 4	D 2 4	L 2 4	選択制御信号 S 2 4
D 5	L 5	カウント出力信号 S 5	D 2 5	L 2 5	DATA, Z, X
D 6	L 6	レクタ出力 S 7, S 2 7	D 2 6	L 2 6	実行終了信号 S 2 6
D 7	L 7	インクリメント出力信号 S 7	D 2 7	L 2 7	分岐制御信号 S 2 7
D 8	アキ		D 2 8	L 2 8	選択制御信号 S 2 8
D 9	L 9	命令信号 S 9	D 2 9	L 2 9	命令実行開始信号 S 2 9
D 1 0	L 1 0	各引数信号 S 1 0	D 3 0	L 3 0	カウント制御信号 S 3 0
D 1 1	L 1 1	DATA	D 3 1	L 3 1	選択制御信号 S 3 1
D 1 2	L 1 2	書込み制御信号 S w	D 3 2	L 3 2	選択制御信号 S 3 2
D 1 3	L 1 3	書込みアドレス A w	D 3 3	L 3 3	読出しアドレス A r
D 1 4	L 1 4	読出しアドレス A r	D 3 4	L 3 4	ラッチ制御信号 S 3 4
D 1 5	L 1 5	外部アドレス	D 3 5	L 3 5	ALU制御信号 S 3 5
D 1 6	L 1 6	外部制御信号 S 1 6	D 3 6	L 3 6	X 値
D 1 7	L 1 7	ラッチ制御信号 S 1 7	D 3 7	L 3 7	書込みアドレス A w
D 1 8	アキ		D 3 8	L 3 8	ラッチ制御信号 S 3 8
D 1 9	アキ		•		
D 2 0	L 2 0	DATA, X, Y	•		
			•		

10/25

FIG. 11

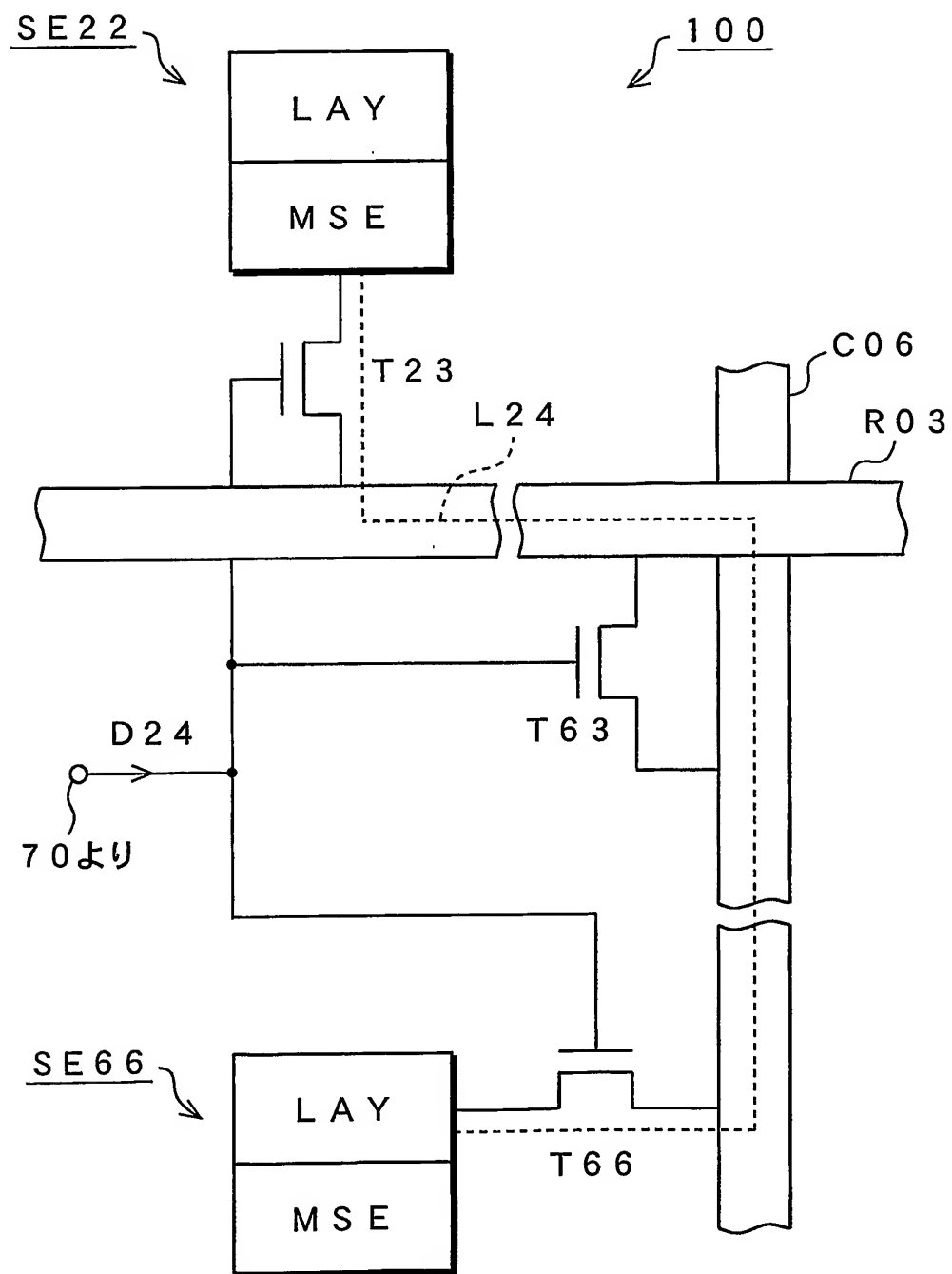
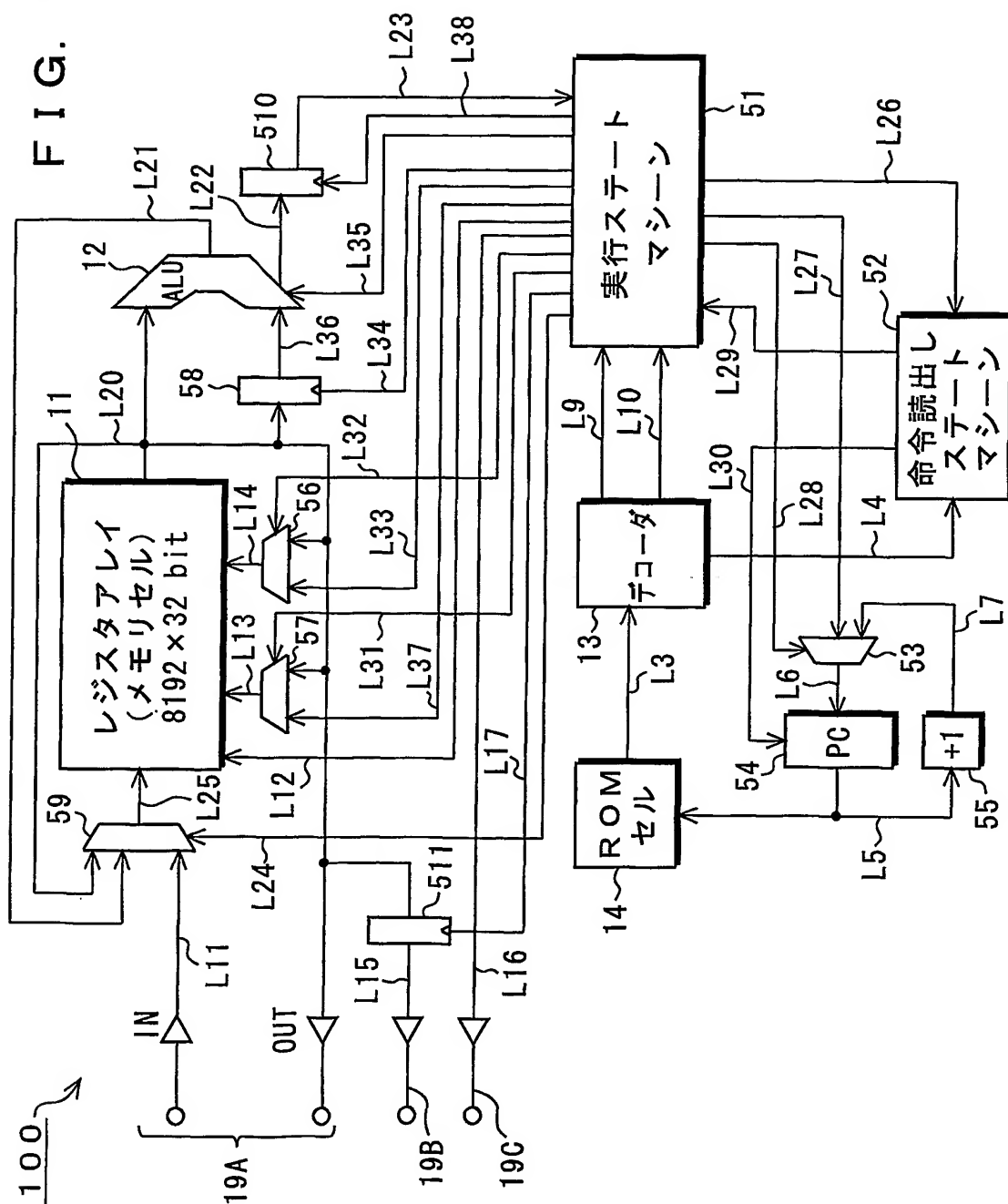
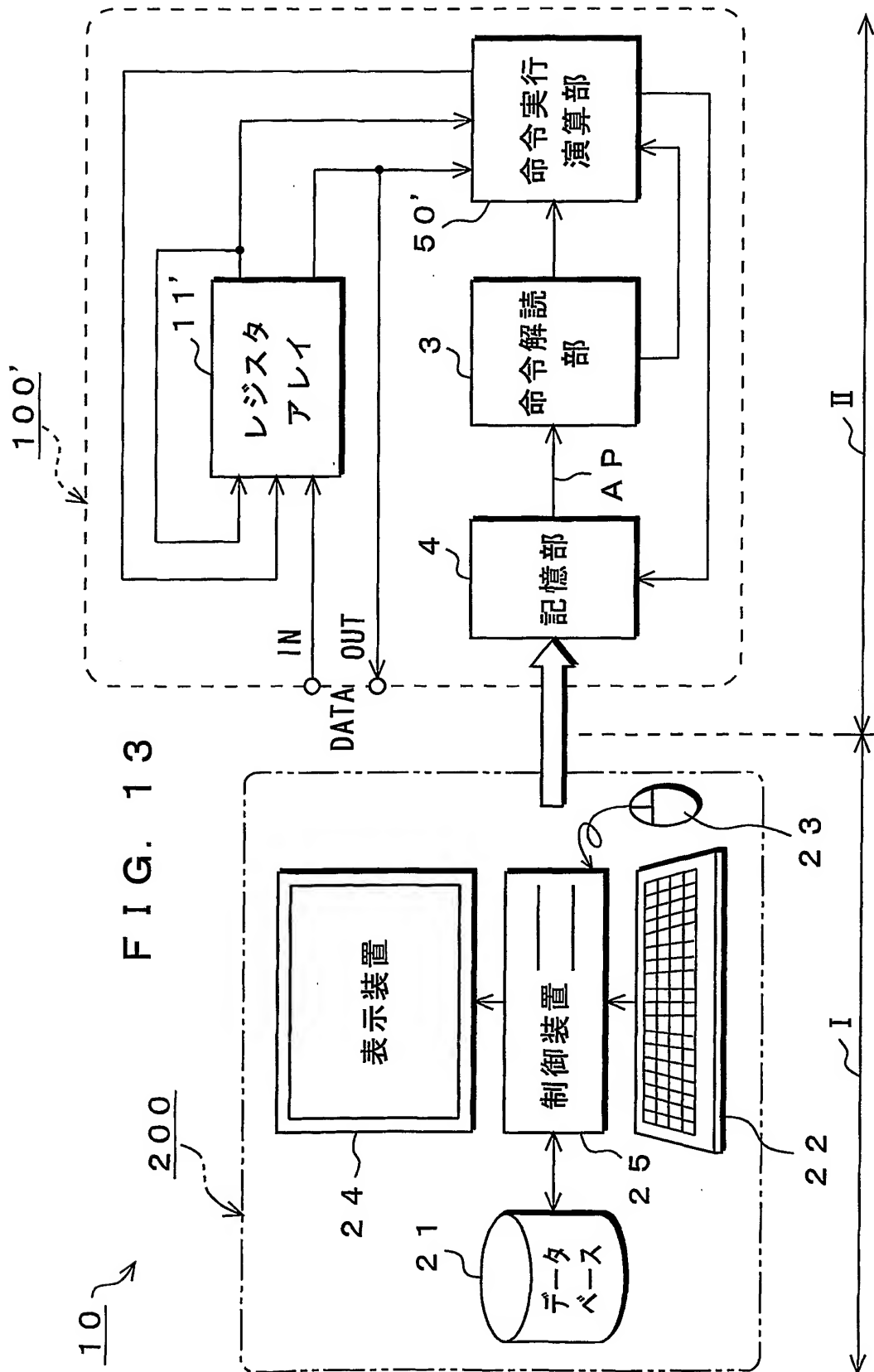


FIG. 12



1 2 / 2 5



1 3 / 2 5

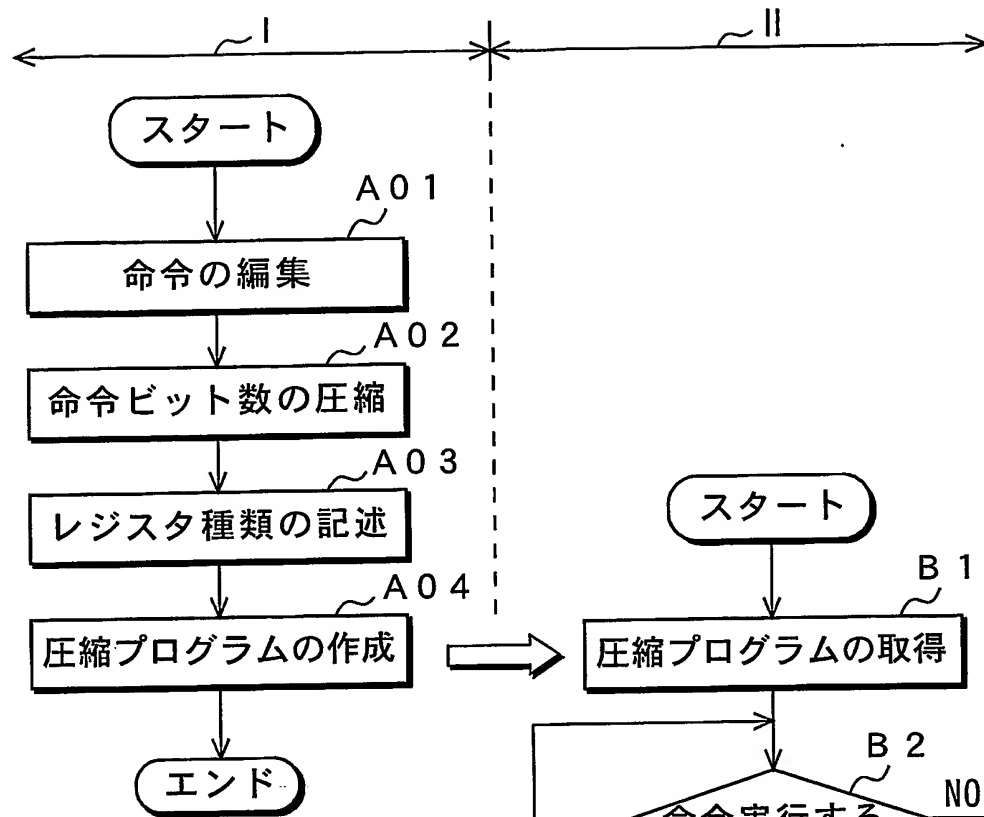


FIG. 14A

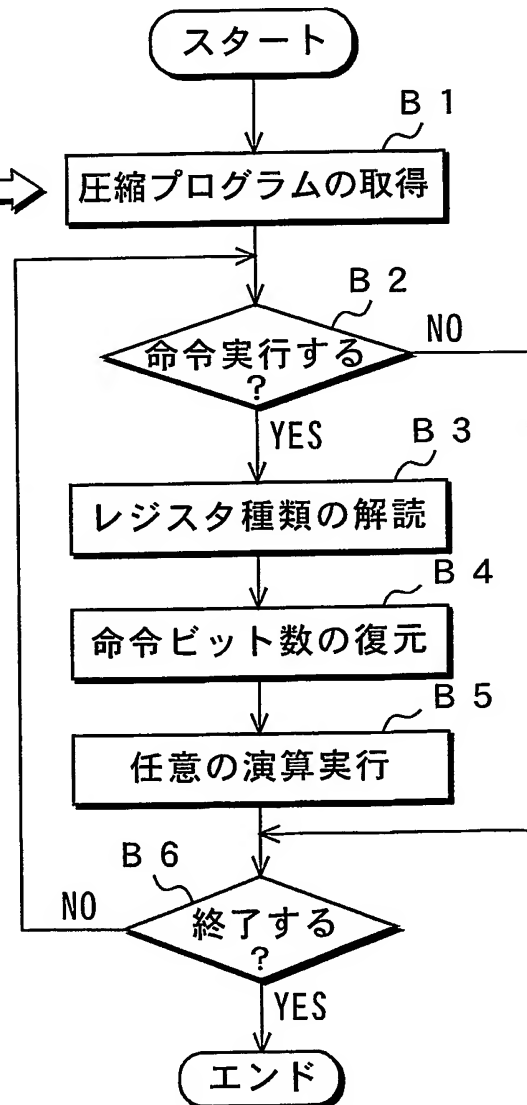


FIG. 14B

1 4 / 2 5

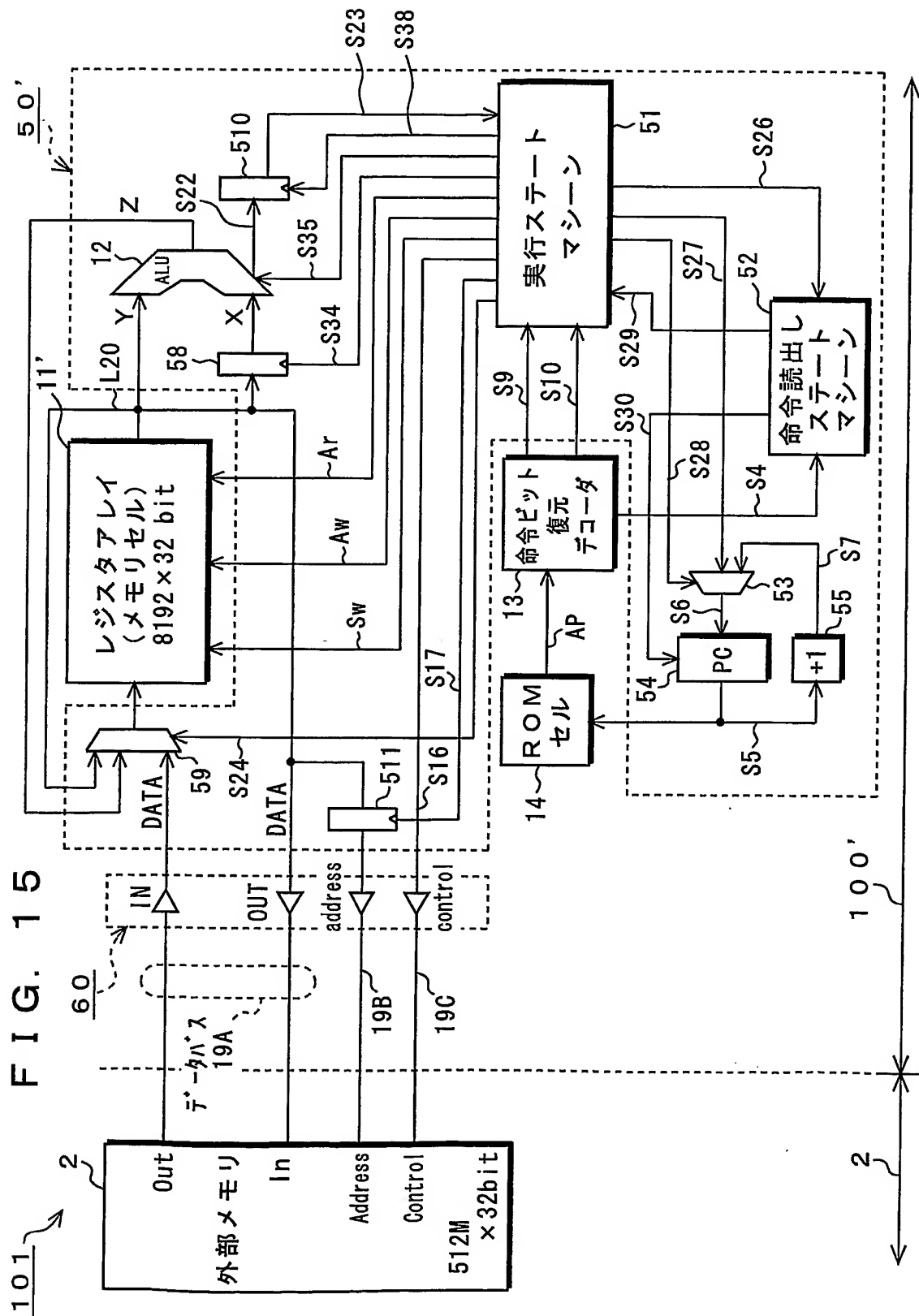
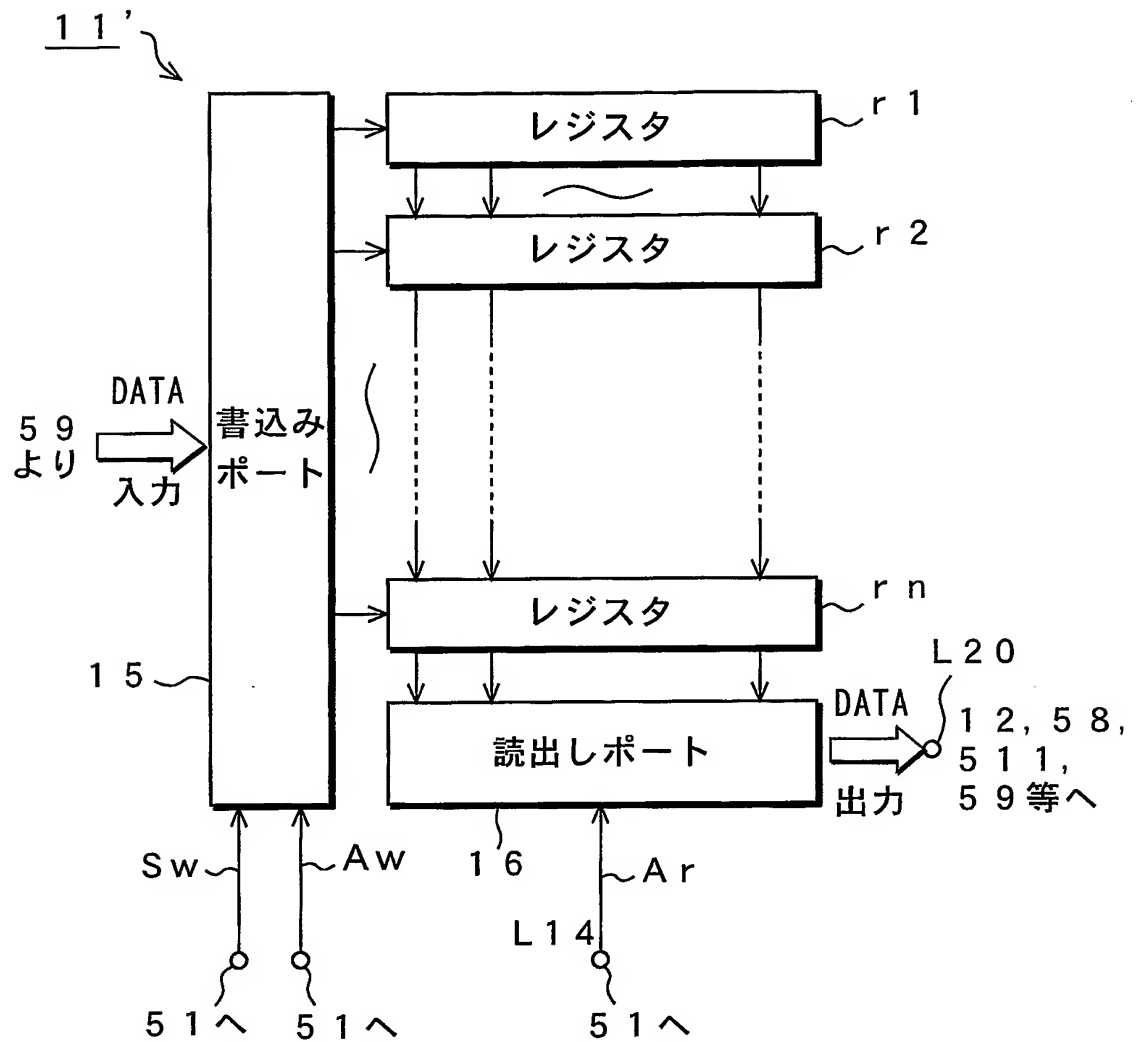
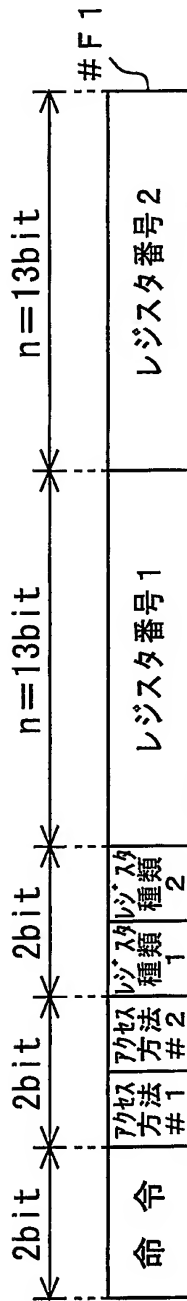


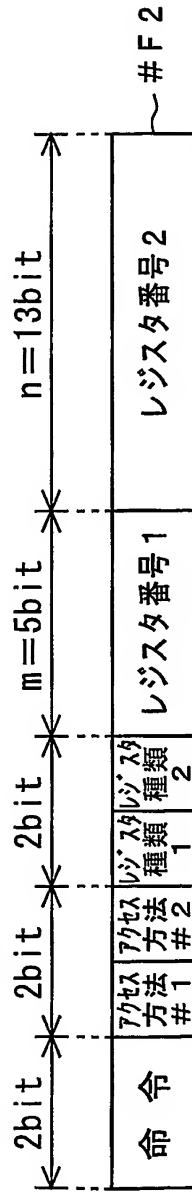
FIG. 16





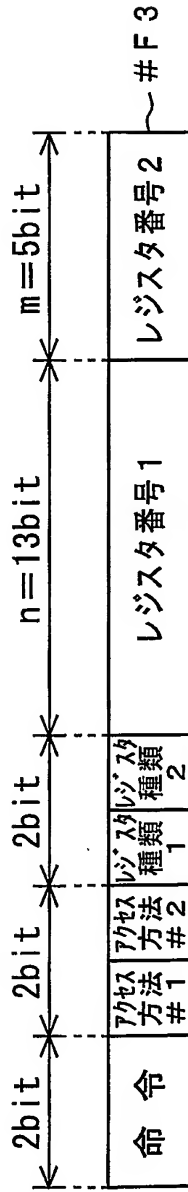
load, add,
cmp 命令

FIG. 17A



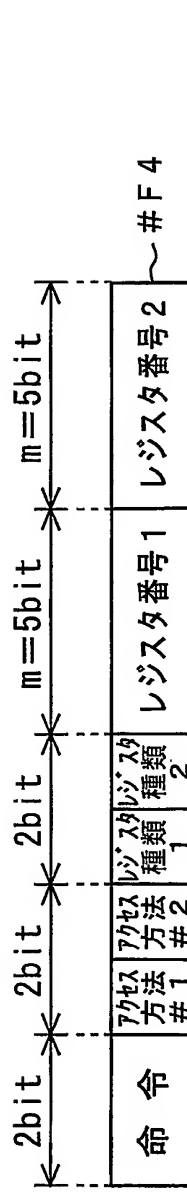
load, add,
cmp 命令

FIG. 17B



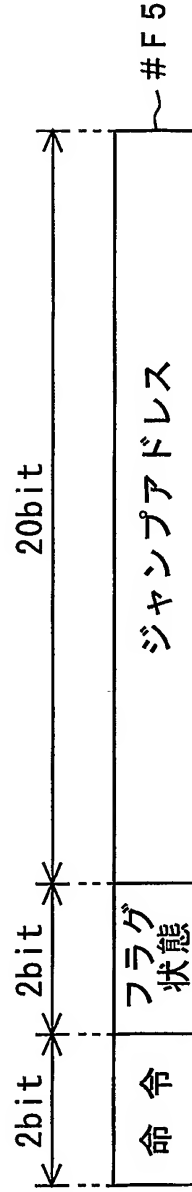
load, add,
cmp 命令

FIG. 17C



load, add,
cmp 命令

FIG. 17D



jump
命令

FIG. 17E

命 令	
0	load
1	add
2	cmp
3	jump

FIG. 18A

アクセス方法 # 1, # 2	
0	レジスタ直接
1	レジスタ相対

FIG. 18B

レジスタ種類 1, 2	
0	レジスタ番号が 31 以下
1	レジスタ番号が 32 以上

FIG. 18C

フラグ状態	
0	無条件
1	zero flag
2	non-zero flag
3	未使用

FIG. 18D

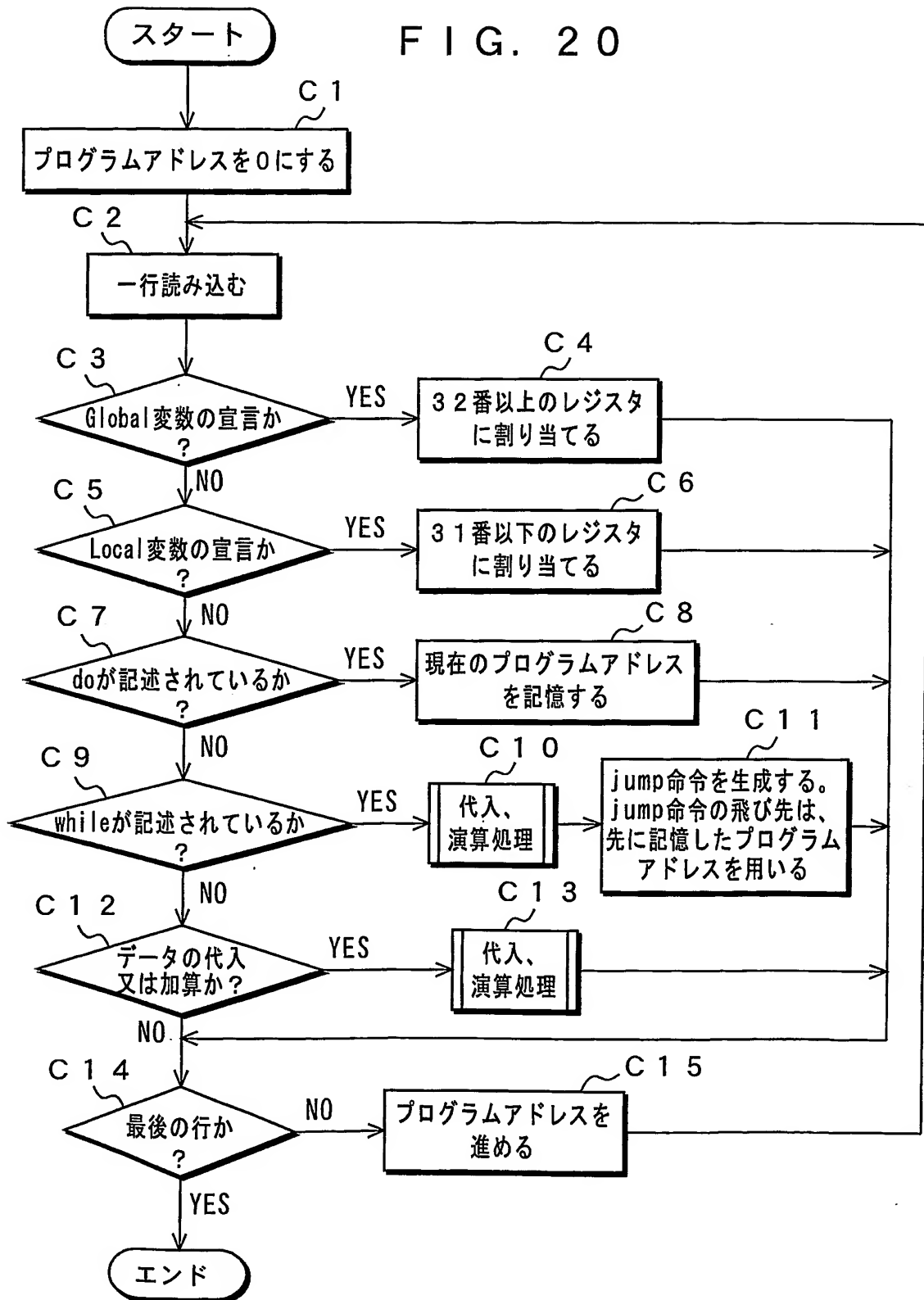
18 / 25

FIG. 19

C言語によるプログラム	記述内容	実施例
<pre> int *read_add, *write_add, counter, end_val; void main() { int temp, added_val; do { temp = *read_add; temp = temp + added_val; *write_add = temp; read_add = read_add + added_val; write_add = write_add + added_val; counter = counter + added_val; } while(counter != end_val); } </pre>	<p>Global変数宣言</p> <p>関数宣言</p> <p>Local変数宣言</p> <p>代入 加算 代入</p> <p>加算 加算 加算</p> <p>比較、分岐</p>	<p>32番目以上のレジスタに割り当てられる</p> <p>31番目以下のレジスタに割り当てられる</p>
P1	P2	P3

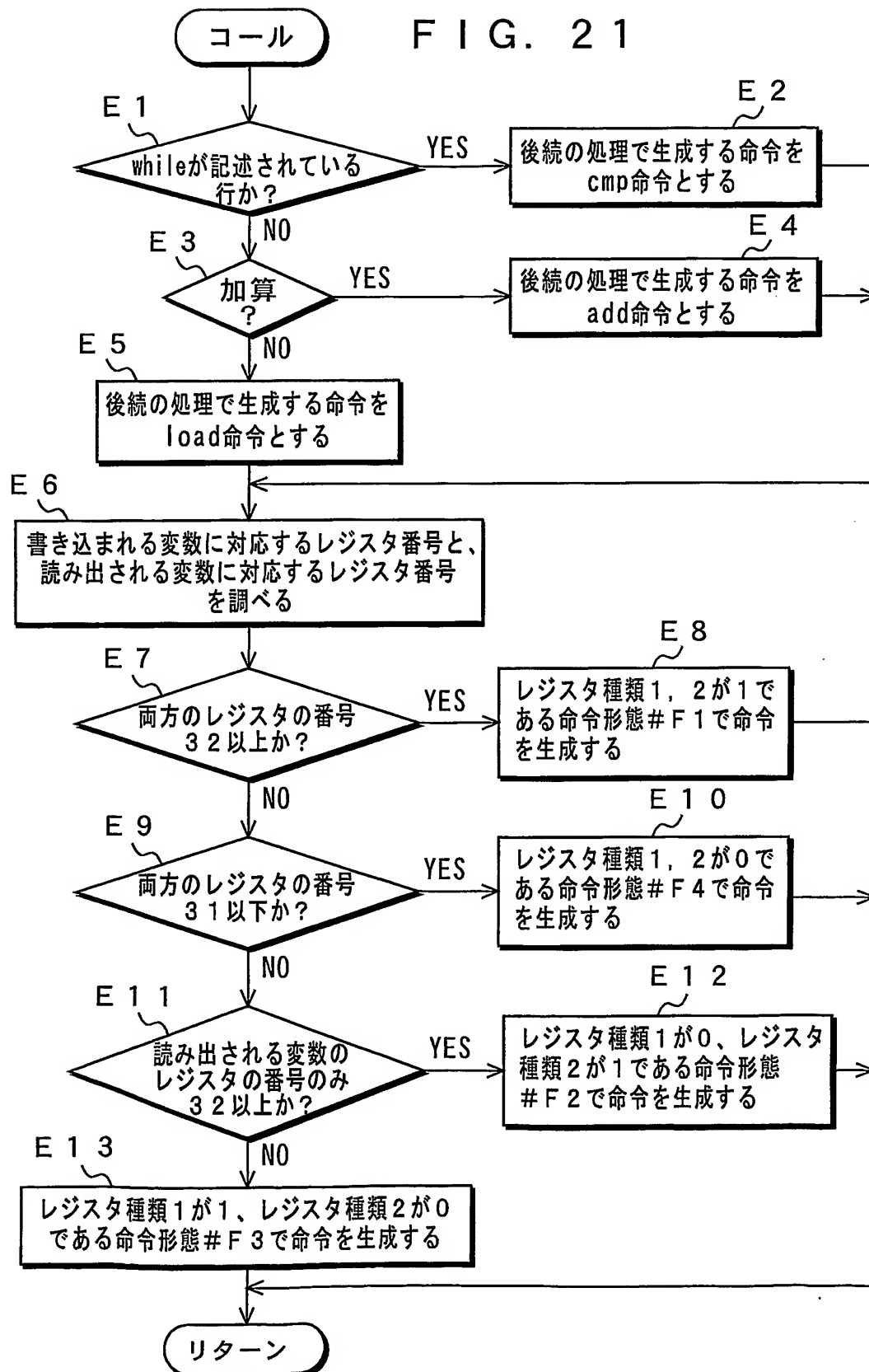
19 / 25

FIG. 20



20 / 25

FIG. 21



21 / 25

FIG. 22

No.	ニーモニックによる表現		機械語による表現
# 1 1	LOOP	load r0, (r32)	140020h
# 1 2		add r0, r1	4001h
# 1 3		load(r33), r0	280420h
# 1 4		add r32, r1	480401h
# 1 5		add r33, r1	480421h
# 1 6		add r34, r1	480441h
# 1 7		cmp r34, r35	8C044023h
# 1 8		jump nz, LOOP	E00000h

FIG. 23

レジスタ番号	値	
r 0	—	← 一時的に使用されるレジスタ
r 1	1	← 加算値
r 3 2	0	← 読み出しアドレス
r 3 3	1 0	← 書き込みアドレス
r 3 4	0	← カウンタ
r 3 5	1 0	← 終了値

22 / 25

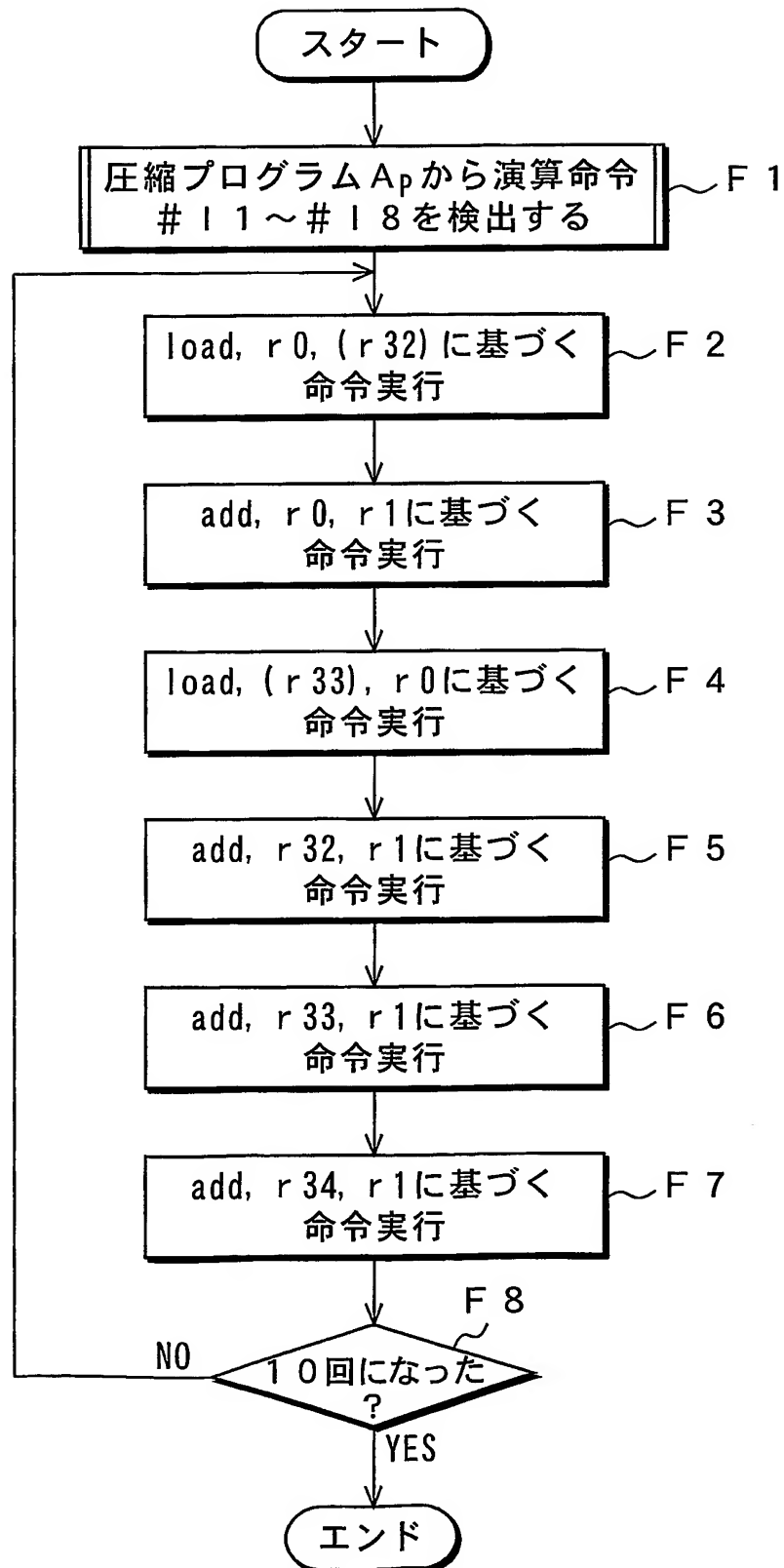
FIG. 24

2 ↘

アドレス	データ	
0000h	0	↑ # M 1 ↓
0001h	0	
0002h	0	
0003h	0	
0004h	0	
0005h	0	
0006h	0	
0007h	0	
0008h	0	
0009h	0	
000Ah	0	↑ # M 2 ↓
000Bh	0	
000Ch	0	
000Dh	0	
000Eh	0	
000Fh	0	
0010h	0	
0011h	0	
0012h	0	
0013h	0	

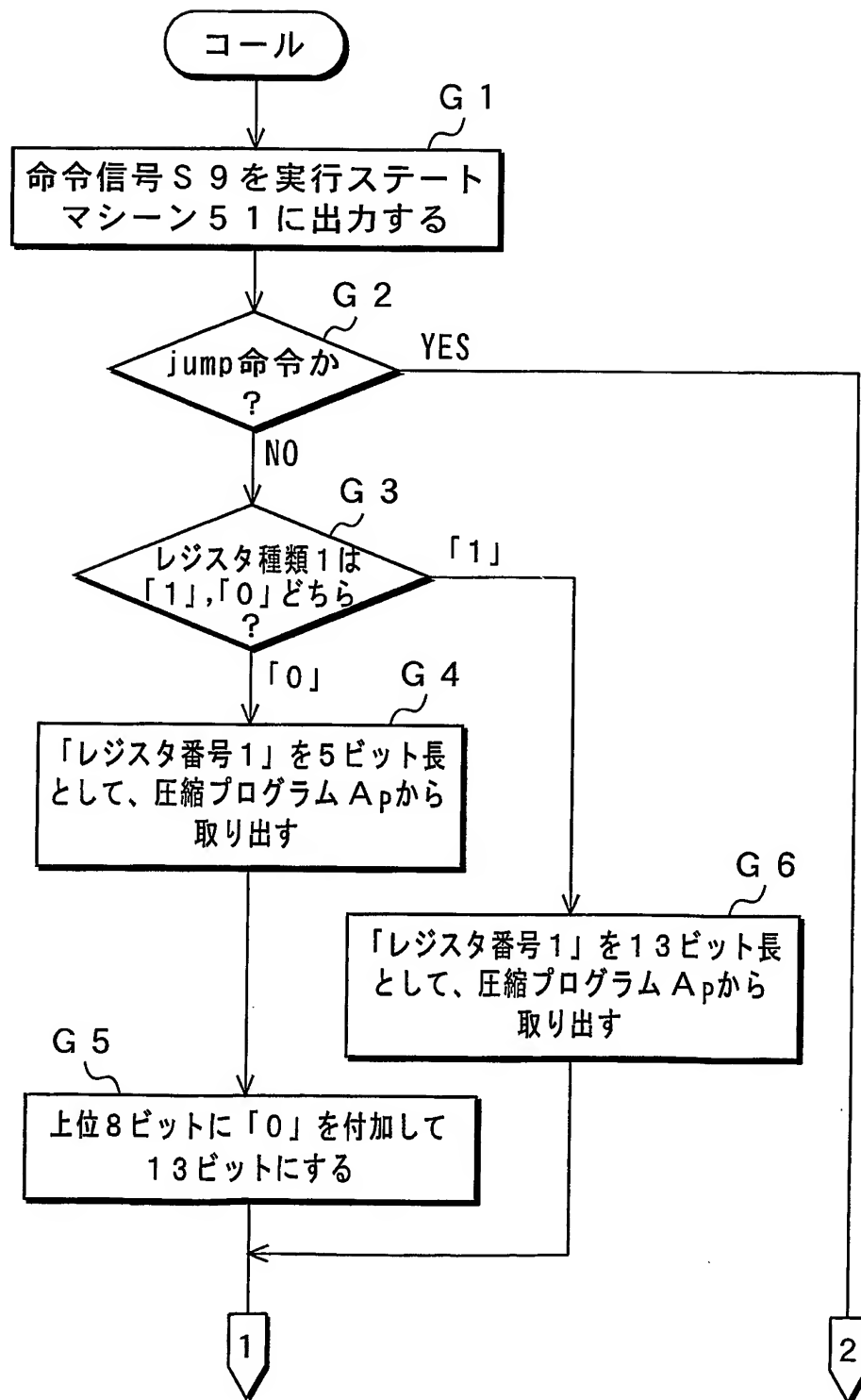
23 / 25

FIG. 25



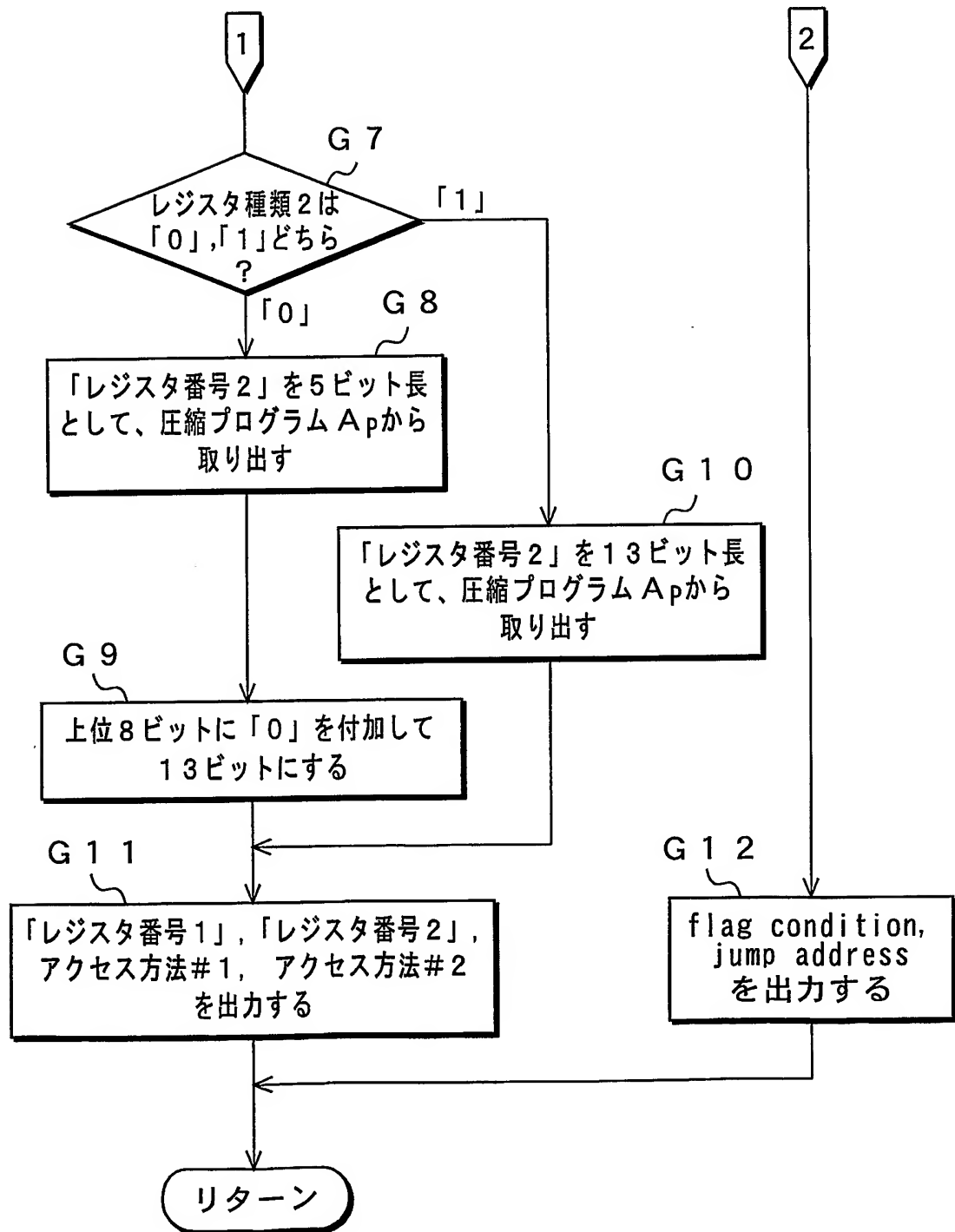
24 / 25

FIG. 26A



25 / 25

FIG. 26B



INTERNATIONAL SEARCH REPORT

 International Application No. .
 PCT/JP03/03716

 A. CLASSIFICATION OF SUBJECT MATTER
 Int.Cl⁷ G06F9/34

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

 Minimum documentation searched (classification system followed by classification symbols)
 Int.Cl⁷ G06F9/30-9/36, G06F15/78

 Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched
 Jitsuyo Shinan Koho 1922-1996 Toroku Jitsuyo Shinan Koho 1994-2003
 Kokai Jitsuyo Shinan Koho 1971-2003 Jitsuyo Shinan Toroku Koho 1996-2003

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X Y	WO 00/19329 A1 (Kabushiki Kaisha Roran), 06 April, 2000 (06.04.00), Full text; all drawings & JP 2000-105759 A	6, 9, 10, 13 8
Y	JP 5-55907 A (Kawasaki Steel Corp.), 05 March, 1993 (05.03.93), Full text; all drawings (Family: none)	8
A	JP 4-91519 A (Hitachi, Ltd. et al.), 25 March, 1992 (25.03.92), Pages 6, lower left column, line 10 to page 7, lower right column, line 13; Figs. 3, 4 (Family: none)	1-25

☒ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

* Special categories of cited documents:
 "A" document defining the general state of the art which is not considered to be of particular relevance
 "E" earlier document but published on or after the international filing date
 "L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
 "O" document referring to an oral disclosure, use, exhibition or other means
 "P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
 "X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
 "Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
 "&" document member of the same patent family

 Date of the actual completion of the international search
 20 June, 2003 (20.06.03)

 Date of mailing of the international search report
 01 July, 2003 (01.07.03)

 Name and mailing address of the ISA/
 Japanese Patent Office

Authorized officer

Facsimile No.

Telephone No.

INTERNATIONAL SEARCH REPORT

International Application No.

PCT/JP03/03716

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	JP 10-320172 A (Seiko Epson Corp.), 04 December, 1998 (04.12.98), Full text; all drawings (Family: none)	17-25

A. 発明の属する分野の分類 (国際特許分類 (IPC))		
Int. Cl ⁷ G06F 9/34		
B. 調査を行った分野		
調査を行った最小限資料 (国際特許分類 (IPC))		
Int. Cl ⁷ G06F 9/30 - 9/36, G06F 15/78		
最小限資料以外の資料で調査を行った分野に含まれるもの 日本国実用新案公報 1922-1996年 日本国公開実用新案公報 1971-2003年 日本国登録実用新案公報 1994-2003年 日本国実用新案登録公報 1996-2003年		
国際調査で使用した電子データベース (データベースの名称、調査に使用した用語)		
C. 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
X	WO 00/19329 A1 (株式会社ローラン) 2000.04.06, 全文, 全図 & JP 2000-105759 A	6, 9, 10, 13
Y		8
Y	JP 5-55907 A (川崎製鉄株式会社) 1993.03.05, 全文, 全図 (ファミリー無し)	8
<input checked="" type="checkbox"/> C欄の続きにも文献が列挙されている。 <input type="checkbox"/> パテントファミリーに関する別紙を参照。		
* 引用文献のカテゴリー 「A」 特に関連のある文献ではなく、一般的技術水準を示すもの 「E」 国際出願日前の出願または特許であるが、国際出願日以後に公表されたもの 「L」 優先権主張に疑義を提起する文献又は他の文献の発行日若しくは他の特別な理由を確立するために引用する文献 (理由を付す) 「O」 口頭による開示、使用、展示等に言及する文献 「P」 国際出願日前で、かつ優先権の主張の基礎となる出願 の日の後に公表された文献 「T」 国際出願日又は優先日後に公表された文献であって出願と矛盾するものではなく、発明の原理又は理論の理解のために引用するもの 「X」 特に関連のある文献であって、当該文献のみで発明の新規性又は進歩性がないと考えられるもの 「Y」 特に関連のある文献であって、当該文献と他の1以上の文献との、当業者にとって自明である組合せによって進歩性がないと考えられるもの 「&」 同一パテントファミリー文献		
国際調査を完了した日 20.06.03		国際調査報告の発送日 01.07.03
国際調査機関の名称及びあて先 日本国特許庁 (ISA/JP) 郵便番号100-8915 東京都千代田区霞が関三丁目4番3号		特許庁審査官 (権限のある職員) 後藤 彰 電話番号 03-3581-1101 内線 3545

C (続き) . 関連すると認められる文献		
引用文献の カテゴリー*	引用文献名 及び一部の箇所が関連するときは、その関連する箇所の表示	関連する 請求の範囲の番号
A	JP 4-91519 A (株式会社日立製作所 外1名) 1992. 03. 25, 第6頁左下欄第10行~第7頁右下欄第13行, 第3図, 第4図 (ファミリー無し)	1-25
A	JP 10-320172 A (セイコーエプソン株式会社) 1998. 12. 04, 全文, 全図 (ファミリー無し)	17-25